




14/JULIOL/2017

MÈTODES DE REPRESENTACIÓ ESPACIAL DE DADES DE POBLACIÓ GEOCODIFICADES

TREBALL FINAL DE MÀSTER

MARC CASTELLÓ

TUTOR IDESCAT: EDUARD SUÑÉ LUÍS / TUTOR UAB: JOAN NUNES
Departament de Geografia, UAB-2017



Índex

ÍNDEX DE FIGURES	- 4 -
ÍNDEX D'SCRIPTS DE SQL.....	- 6 -
ÍNDEX D'SCRIPTS DE JAVA.....	- 7 -
ABSTRACT	- 8 -
AGRAÏMENTS	- 9 -
1.-INTRODUCCIÓ.....	- 10 -
1.1.-MARC INSTITUCIONAL.....	- 10 -
1.2.-REGISTRE DE LA POBLACIÓ D'IDECAT	- 10 -
1.3.-REPRESENTACIÓ ESPACIAL DEL REGISTRE DE LA POBLACIÓ	- 11 -
1.4.-MÈTODES DE REPRESENTACIÓ ESPACIAL.....	- 11 -
1.4.1.-Representació mitjançant quadrees.....	- 12 -
1.4.2.-Representació mitjançant la pertorbació de coordenades	- 12 -
2.-OBJECTIUS DEL TREBALL	- 13 -
2.1.-OBJECTIUS.....	- 13 -
2.2.-ABAST	- 13 -
2.3.-PLA DE TREBALL	- 13 -
3.-MARC TEÒRIC	- 15 -
3.1.-QUADTREES.....	- 15 -
3.1.1-Paràmetres dels quadrees.....	- 17 -
3.1.1.1.- L'indar de població	- 17 -
3.1.1.2-Resolució màxima.....	- 19 -
3.1.1.3-Resolució mínima	- 19 -
3.2.-MÈTODE DE MONTECARLO PER A L'ESTIMACIÓ DE L'ERROR.	- 19 -
3.3.-PERTORBACIÓ DE COORDENADES	- 20 -
4.-DADES D'IDECAT	- 22 -
4.1.-REGISTRE DE LA POBLACIÓ	- 22 -
4.2.-GEOMETRIA DELS QUADTREES	- 22 -
4.3.-POLÍGONS DE MONTECARLO	- 23 -
5.-METODOLOGIA.....	- 24 -
5.1.-METODOLOGIA DE CREACIÓ I VALIDACIÓ DELS QUADTREES	- 24 -
5.1.1.-Creació dels quadrees	- 24 -
5.1.3.-Avaluació de l'error dels quadrees	- 25 -
5.2.-METODOLOGIA DE LA PERTORBACIÓ DE COORDENADES I LA SEVA VALIDACIÓ.....	- 26 -
5.2.1.- Creació de l'espai de publicació i l'espai de comparació	- 26 -
5.2.2.-Funció d'intercanvi de coordenades	- 27 -
5.2.3.- Avaluació de l'error de la pertorbació de coordenades	- 28 -
6.- PREPARACIÓ DE LES DADES I OBTENCIÓ DELS PARÀMETRES PREVIS A LA CREACIÓ DELS QUADTREES	- 29 -
6.1.-DETERMINACIÓ DELS LLINDARS MÉS ADIENTS.....	- 29 -
6.1.1-Creació de la taula del Registre de Població geocodificat	- 29 -
6.1.2.-Intersecció de les dades de la població amb la geometria del quadtree	- 29 -
6.1.3.-Anàlisi de freqüències de la població segons la geometria a diferents nivells	- 30 -
6.2.-DETERMINACIÓ DE LA RESOLUCIÓ MÀXIMA.....	- 31 -

6.2.1.-Creació de nova geometria.....	- 31 -
6.2.2.-Creació de la taula on s'agrega la nova geometria.....	- 32 -
6.2.3.-Divisió de la geometria a partir de la funció DIVIDE.....	- 32 -
6.2.4.-Retall de la geometria segons el contorn de Catalunya	- 35 -
6.2.5.-Inserció de la nova geometria a la geometria del quadtree.....	- 35 -
6.3.- Determinació de la resolució mínima	- 37 -
7.-IMPLEMENTACIÓ DELS MÈTODES DE QUADTREES I DE LA PERTORBACIÓ DE COORDENADES	- 38 -
7.1.-IMPLEMENTACIÓ DELS QUADTREES	- 38 -
7.1.1.-Procediment de creació de la informació agregada de la població.....	- 38 -
7.1.1.1.-Creació de la taula dels agregats	- 39 -
7.1.1.2.-Omplir la taula d'agregats amb la resolució màxima	- 39 -
7.1.1.3.-Omplir la taula d'agregats recursivament	- 39 -
7.1.1.4.-Encadenament de funcions per a calcular els agregats de la població.....	- 40 -
7.1.2.-Procediment de creació del quadtree	- 40 -
7.1.2.1.-Creació de la taula del quadtree	- 41 -
7.1.2.2.-Anàlisi de cada element geomètric	- 42 -
7.1.2.3.-Anàlisi dels 4 fills de l'element geomètric analitzat	- 42 -
7.1.2.4.-Inserció dels elements geomètrics a la taula.....	- 42 -
7.1.2.5.-Encadenament de les funcions per a crear el quadtree	- 43 -
7.1.3.-Inserció de la informació agregada de la població i de la geometria del quadtree creat ...	- 43 -
7.1.4.-Característiques dels quadrees creats.....	- 43 -
7.2.-IMPLEMENTACIÓ DE LA PERTORBACIÓ DE COORDENADES	- 44 -
7.2.1.-Espai de comparació i espai de publicació.....	- 44 -
7.2.2.-Implementació de la matriu d'intercanvi	- 45 -
7.2.3.-Creació de la taula d'habitants amb la pertorbació de coordenades.....	- 46 -
8.-CÀLCUL DELS ERRORS	- 47 -
8.1.-POLÍGONS DE MONTECARLO	- 47 -
8.2.-CÀLCUL DELS ERRORS EN ELS QUADTREE CREATS	- 47 -
8.2.1.-Creació de la taula dels experiments de Montecarlo.....	- 47 -
8.2.2.-Càlcul de la població exacta dels polígons.....	- 48 -
8.2.3.-Càlcul de la població segons cada quadtree.....	- 48 -
8.2.4.-Càlcul dels errors	- 49 -
8.2.5.-Distribució de l'error relatiu en els experiments	- 49 -
8.2.6.-Distribució de les diferències relatives.....	- 52 -
8.2.7.-L'error relatiu segons l'àrea del polígon	- 54 -
8.3.-ERRORS DE LA PERTORBACIÓ DE COORDENADES	- 57 -
8.3.1.- Anàlisi de khi quadrat	- 57 -
9.-CONCLUSIONS	- 59 -
9.1.-CONCLUSIONS SOBRE ELS OBJECTIUS	- 59 -
9.1.1.-Paràmetres del quadtree	- 59 -
9.1.1.1.-Resolució màxima	- 59 -
9.1.1.2.-Resolució mínima	- 59 -
9.1.1.3.-Llindar	- 60 -
9.1.2.-El quadtree idoni.....	- 61 -
9.1.2.- Pertorbació de coordenades	- 62 -
9.2.-CONCLUSIONS SOBRE LA METODOLOGIA	- 62 -
9.3.-CONCLUSIONS SOBRE FUTURES APLICACIONS DE LES DADES	- 63 -
10.-REFERÈNCIES.....	- 64 -
11.-ANNEXOS.....	- 65 -

ÍNDIX DE FIGURES

FIGURA 1 BASE DE DADES ASSOCIADA AL REGISTRE D'HABITANTS (SUÑÉ, 2015)	- 11 -
FIGURA 2 PLA DE TREBALL	- 14 -
FIGURA 3 CALENDARI DELS OBJECTIUS	- 14 -
FIGURA 4 EXEMPLE DE DIVISIÓ RECURSIVA DE L'ESPAI (LAGONIGRO RAYMOND ET AL, 2016)	- 15 -
FIGURA 5 EXEMPLE DE DIVISIÓ RECURSIVA DE L'ESPAI II (LAGONIGRO RAYMOND ET AL, 2016)	- 15 -
FIGURA 6 PROCÉS DE TRANSFORMACIÓ DE DADES DEMOGRÀFIQUES A QUADTREE (MARTIN BEHNISCH ET AL, 2013)	- 16 -
FIGURA 7 EXEMPLE DE QUADTREE QUE CONTÉ INFORMACIÓ DEMOGRÀFICA (LAGONIGRO RAYMOND ET AL, 2016)	- 17 -
FIGURA 8 CREACIÓ DEL QUADTREE A PARTIR DE L'AGREGACIÓ ESPACIAL AMB UN LLINDAR DE 10 (LAGONIGRO RAYMOND ET AL, 2016)	- 18 -
FIGURA 9 DETALL DE LA CREACIÓ DEL QUADTREE A PARTIR DE L'AGREGACIÓ ESPACIAL AMB UN LLINDAR DE 10 FONT: (LAGONIGRO RAYMOND ET AL, 2016)	- 19 -
FIGURA 10 ESQUEMA D'INTERCANVI (SWITCH) DE COORDENADES	- 21 -
FIGURA 11 DIAGRAMA DE CREACIÓ DELS QUADTREES	- 24 -
FIGURA 12 DIAGRAMA II DE CREACIÓ DELS QUADTREES	- 25 -
FIGURA 13 FUNCIÓ D'INTERCANVI DE COORDENADES ENTRE ELS PUNTS A I B	- 27 -
FIGURA 14 REPRESENTACIÓ LINEAL DE LA SUMA DE LA FREQUÈNCIA ACUMULADA DE LA POBLACIÓ	- 30 -
FIGURA 15 EXEMPLE DE LA GEOMETRIA D'1KM AL DELTA DE L'ÈBRE	- 32 -
FIGURA 16 INICI DEL SEGON BUCLE DE LA FUNCIÓ DIVIDE	- 33 -
FIGURA 17 SEGON PAS DEL SEGON BUCLE DE LA FUNCIÓ DIVIDE	- 33 -
FIGURA 18 PRIMER PAS DEL SEGON BUCLE DE LA FUNCIÓ DIVIDE AMB LA Y CANVIADA	- 34 -
FIGURA 19 SEGON PAS DEL SEGON BUCLE DE LA FUNCIÓ DIVIDE AMB LA Y CANVIADA	- 34 -
FIGURA 20 COLUMNA GRD_FIXID DE LA TAULA GRID_ETRS89_LALEA_MULTIRESOL, ON HI HA LES COORDENADES X I Y ORIGINALS	- 34 -
FIGURA 21 VISTA DE BARCELONA AMB LA GEOMETRIA DE MALLA DE 62.5M I LA POBLACIÓ AGREGADA PER SOBRE DEL LLINDAR DE POBLACIÓ 17	- 36 -
FIGURA 22 VISTA II DE BARCELONA AMB LA GEOMETRIA DE MALLA DE 62.5 M I LA POBLACIÓ AGREGADA PER SOBRE DEL LLINDAR DE POBLACIÓ 17	- 36 -
FIGURA 23 VISTA III DE BARCELONA AMB LA GEOMETRIA DE MALLA DE 62.5 M I LA POBLACIÓ AGREGADA PER SOBRE DEL LLINDAR DE POBLACIÓ 17	- 37 -
FIGURA 24 DIAGRAMA DEL PROCEDIMENT DE CREACIÓ DE LA INFORMACIÓ AGREGADA DE LA POBLACIÓ	- 38 -
FIGURA 25 RESUM DELS QUADTREES CREATS	- 41 -
FIGURA 26 DIAGRAMA DEL PROCEDIMENT DE CREACIÓ DEL QUADTREE	- 41 -
FIGURA 27 NOMBRE DE QUADRATS CALCULAT PER A CADA RESOLUCIÓ	- 44 -
FIGURA 28 DIAGRAMA D'IMPLEMENTACIÓ DE LA PERTORBACIÓ DE COORDENADES	- 44 -
FIGURA 29 DIAGRAMA DEL CÀLCUL DELS ERRORS EN ELS QUADTREES CREATS	- 47 -
FIGURA 30 DIAGRAMA DEL CÀLCUL DE POBLACIÓ EXACTA ENS ELS POLÍGONS DE MONTECARLO	- 48 -
FIGURA 31 DIAGRAMA DEL CÀLCUL DE POBLACIÓ SEGONS CADA QUADTREE	- 49 -
FIGURA 32 DIAGRAMES DE CAIXA DELS DIFERENTS QUADTREES	- 50 -
FIGURA 33 DISTRIBUCIÓ DE L'ERROR (MEDIANA) RESPECTE DEL LLINDAR AMB LA RESOLUCIÓ MÍNIMA	- 51 -
FIGURA 34 DISTRIBUCIÓ DE L'ERROR (MITJANA) RESPECTE DEL LLINDAR AMB LA RESOLUCIÓ MÍNIMA	- 52 -
FIGURA 35 HISTOGRAMA DE LA DISTRIBUCIÓ DE LES DIFERÈNCIES RELATIVES	- 53 -
FIGURA 36 DIAGRAMES DE DISPERSIÓ DE L'ERROR RELATIU EN RELACIÓ A L'ÀREA DEL POLÍGON AMB LLINDAR 17	- 55 -
FIGURA 37 ERROR RELATIU D'UN CAS ATÍPIC SEGONS EL QUADTREE {62.5M,62.5M,RP14,17}	- 56 -
FIGURA 38 ERROR RELATIU DEL CAS ATÍPIC ANTERIOR SEGONS EL QUADTREE {62.5M,62.5M,RP14,17}	- 56 -
FIGURA 39 ERRORS RELATIUS EXTREMS INFERIORS PER AL QUADTREE {62.5M,250M,RP2014,17}	- 57 -
FIGURA 40 DISTÀNCIA DE KHI QUADRAT SOBRE ELS VECTORS NORMALITZATS	- 57 -
FIGURA 41 DIAGRAMA DE DISPERSIÓ ENTRE LA DISTÀNCIA DE KHI QUADRAT I LA POBLACIÓ	- 58 -

FIGURA 42 Taula de comparació de la població nul•la en els diferents quadrees amb llindar 17	- 60 -
FIGURA 43 Taula resum del llindar en els quadrees 62,5_250_RP2014	- 60 -
FIGURA 44 Taula resum del recompte de quadrats i de quadrats nuls	- 61 -
FIGURA 45 Taula de la població nul•la en el quadree 62,5_250_RP2014_17	- 61 -

ÍNDEX D'SCRIPTS DE SQL

SCRIPT DE SQL 1 CREACIÓ DE LA TAULA HABITANTS_EDAT_POSICIO	- 65 -
SCRIPT DE SQL 2 INTERSECCIÓ DE LES DADES DE POBLACIÓ AMB LES GEOMETRIES DE TOTS ELS NIVELLS DEL QUADTREE	- 66 -
SCRIPT DE SQL 3 CÀLCUL DE LES FREQÜÈNCIES DE LA POBLACIÓ EN LA RESOLUCIÓ ESPACIAL DE 1000M	- 67 -
SCRIPT DE SQL 4 FUNCIO DIVIDE	- 68 -
SCRIPT DE SQL 5 FUNCIO PER OBTENIR LA COORDENADA X	- 70 -
SCRIPT DE SQL 6 FUNCIO PER OBTENIR LA COORDENADA Y	- 71 -
SCRIPT DE SQL 7 EXECUCIÓ DELES FUNCIONS DIVIDE,GETX_FROM_GRD_FIXID I GETY_FROM_GRD_FIXID	- 72 -
SCRIPT DE SQL 8 INTERSECCIÓ DELS QUADRATS DE 62.5M AMB EL LÍMIT DE CATALUNYA	- 73 -
SCRIPT DE SQL 9 ELIMINACIÓ DE LA GEOMETRIA DE 62.5M QUE ESTÀ FORA DEL LÍMIT DE CATALUNYA	- 74 -
SCRIPT DE SQL 10 INSERCIÓ DE LA NOVA GEOMETRIA A LA TAULA DE LA GEOMETRIA	- 75 -
SCRIPT DE SQL 11 UNIÓ DE LA TAULA DE LA GEOMETRIA AMB LA TAULA ON HI HA LA RELACIÓ RECURSIVA	- 76 -
SCRIPT DE SQL 12 COMPROVACIÓ DE LA RECURSIVITAT EN EL NIVELL 4	- 77 -
SCRIPT DE SQL 13 CANVI DE LA GEOMETRIA ORIGINAL A LA GEOMETRIA (ETRS89_UTM31N)	- 78 -
SCRIPT DE SQL 14 FUNCIO CREA_TAULA_AGREGATS	- 79 -
SCRIPT DE SQL 15 FUNCIO OMPLE_AGG_MAX_RESOLUCIO	- 80 -
SCRIPT DE SQL 16 FUNCIO AGREGA_EN_JERARQUIA	- 81 -
SCRIPT DE SQL 17 FUNCIO CREACIO_QUADTREE_INFO	- 82 -
SCRIPT DE SQL 18 FUNCIO CREA_TAULA_QUADTREE	- 83 -
SCRIPT DE SQL 19 FUNCIO ANALIZE_ELEMENT	- 84 -
SCRIPT DE SQL 20 FUNCIO MUSTAGGREGATE	- 85 -
SCRIPT DE SQL 21 FUNCIO INSERTA_ELEMENT	- 86 -
SCRIPT DE SQL 22 FUNCIO CREA_QUADTREE	- 87 -
SCRIPT DE SQL 23 REUNIÓ DE LA INFORMACIÓ AGREGADA DE LA POBLACIÓ I DE LA GEOMETRIA DE CADA ELEMENT DEL QUADTREE, EN EL CAS DEL QUADTREE QT_4_3_HABITANTS_EDAT_POSICIO_13	- 88 -
SCRIPT DE SQL 24 FUNCIO DE CREACIÓ DE L'ESPAI DE COMPARACIÓ	- 89 -
SCRIPT DE SQL 25 FUNCIO DE CREACIÓ DE L'ESPAI DE MESURA	- 90 -
SCRIPT DE SQL 26 CREACIÓ DE LA TAULA DELS EXPERIMENTS DE MONTECARLO	- 91 -
SCRIPT DE SQL 27 CÀLCUL DE LA POBLACIÓ EXACTA ENS ELS POLÍGONS DE MONTECARLO	- 92 -
SCRIPT DE SQL 28 CÀLCUL DE LA POBLACIÓ SEGONS CADA QUADTREE	- 93 -
SCRIPT DE SQL 29 CÀLCUL DE L'ERROR RELATIU EN ELS EXPERIMENTS	- 94 -
SCRIPT DE SQL 30 CÀLCUL DE LES DIFERÈNCIES RELATIVES	- 95 -
SCRIPT DE SQL 31 CREACIÓ DEL CODI DE DISTRIBUCIÓ DE LES EDATS, VECTOR DE DIMENSIÓ 19	- 96 -

ÍNDIX D'SCRIPTS DE JAVA

SCRIPT DE JAVA 1 FÓRMULA DE LA FUNCIO DE PERTORBACIO DE COORDENADES.....	- 97 -
SCRIPT DE JAVA 2 FÓRMULA DE CREACIO DE LA MATRIU D'INTERCANVI	- 99 -
SCRIPT DE JAVA 3 EMPLERAMENT DE LA TAULA D' HABITANTS_EDAT_POSICIO_PERTURBAT	- 103 -

Abstract

La informació de la població georeferenciada posa en risc el secret estadístic dels habitants, però alhora aporta grans possibilitats d'estudis i d'anàlisis en detall degut a la molt alta escala espacial resultant. Davant d'aquesta situació, el present treball mostra els quadrees i la perturbació de coordenades com a mètodes per la representació espacial de la població amb l'objectiu de trobar l'equilibri entre el secret estadístic i els avantatges de la georeferenciació.

En aquest treball la informació de la població és el Registre de la Població de Catalunya de l'any 2014. En ambdós mètodes de representació espacial es presenta un breu resum dels seus respectius cossos teòrics, la metodologia d'aplicació a la representació espacial de la població, la implementació amb els corresponents scripts i l'anàlisi d'errors respecte a la població georeferenciada. L'anàlisi d'errors s'ha dut a terme gràcies al mètode de Montecarlo, emprat en aquest cas per a la generació d'àrees d'interès aleatòries.

La información de la población georeferenciada pone en riesgo el secreto estadístico de los habitantes, pero a la vez conlleva grandes posibilidades de estudios y análisis en detalle debido a la muy alta escala espacial resultante. Frente a esta situación, este trabajo muestra los quadrees y la perturbación de coordenadas como métodos para la representación espacial de la población con el objetivo de encontrar un equilibrio entre el secreto estadístico y las ventajas de la georeferenciación.

En este trabajo la información de la población es el Registro de la Población de Catalunya del año 2014. En ambos métodos de representación espacial se presenta un breve resumen de sus respectivos marcos teóricos, la metodología de aplicación a la representación espacial de la población, la implementación con los correspondientes scripts y el análisis de errores respecto a la población georeferenciada. El análisis de errores se ha llevado a cabo gracias al método de Montecarlo, empleado en este caso para la generación de áreas de interés aleatorias.

The information on georeferenced population puts at risk the statistical secret of the inhabitants, but on the other hand it greatly enlarges the possibilities of studies and analysis in detail due to the very high spatial scale obtained. In front of this situation, this project shows the quadrees and the perturbation of coordinates as methods for the spatial representation of the population with the aim to find a balance between the statistical secret and the advantages of georeferencing.

In this project, the population's information is the Register of the Population of Catalonia of the year 2014. For both methods of spatial representation, it is shown a brief summary of its theoretical bodies, the methodology to apply them to the spatial representation of the population, the implementation with their corresponding scripts and the analysis of errors with respect to the exact georeferenced population. The analysis of errors has been performed using the MonteCarlo method, in this case to generate random areas of interest.

Agraïments

Voldria expressar el meu agraïment a totes aquelles persones que han contribuït a la realització d'aquest treball de final de màster:

En primer lloc agraeixo la feina, els consells, l'ajuda i dedicació durant tot el desenvolupament de les pràctiques a Idescat i en la realització d'aquest treball al tutor Eduard Suñe. Sense ell hauria estat impossible realitzar aquest treball.

En segon lloc agraeixo a Idescat per oferir-me la possibilitat de realitzar les pràctiques amb ells. Ha estat una gran experiència acadèmica, professional i personal molt gran i satisfactòria.

En tercer lloc agraeixo l'ajuda del meu tutor de la UAB, Joan Nunes, perquè sempre em va donar un cop de mà quan més ho necessitava.

Per últim, agreixo de tot cor el suport de la família, parella, amics i amigues per la comprensió, confiança i ajuda que m'han proporcionat al llarg de tot el període de pràctiques i en la confecció d'aquest treball.

1.-Introducció

Aquest treball s'emmarca en el Treball Final del Màster en Tecnologies de la Informació Geogràfica 19a.edició,el qual es realitza mitjançant pràctiques externes en una entitat col·laboradora. Aquestes pràctiques s'han realitzat a l'Institut d'Estadística de Catalunya (Idescat).

1.1.-Marc institucional

Idescat està treballant sobre els mètodes de representació espacial de la informació geocodificada, en concret, la del Registre de la Població (RP). Aquesta representació espacial, comporta una sèrie de reptes, ja que la informació és especialment sensible a causa del secret estadístic, però al mateix temps, Idescat vol representar aquesta informació amb la màxima resolució espacial i precisió possible.

Dintre d'aquest context, el present treball explora alguns dels diferents mètodes de representació espacial existents i analitza quins són els paràmetres que permeten una millor precisió espacial d'aquestes dades sense que aquestes vulnerin el secret estadístic.

Aquestes treball és la materialització d'aquestes pràctiques que s'han realitzat en un total de 250 hores al llarg de l'Abril, Maig i Juny de l'any 2017.

1.2.-Registre de la Població d'Idescat

Idescat està treballant des de l'any 2012, en un sistema d'informació estadística basat en els registres administratius. Concretament, en el cas de la població, el Registre de la Població (RP). Aquest registre és un producte derivat del Padró d'Habitants, gestionat per l'Instituto Nacional de Estadística (INE), que periòdicament facilita la informació a Idescat.

El Registre de la Població conté informació demogràfica: sexe, edat, lloc de residència, nacionalitat i també informació territorial-administrativa; província, municipi, districte, secció censal i adreça postal.

L'adreça postal està descrita per una sèrie de camps com són el tipus de via, codi de via, nom de la via i uns camps de numeració que descriuen la part horitzontal i vertical de l'adreça postal. Un primer tractament, que agrupa els habitants que resideixen en una mateixa adreça permet obtenir la base de dades del tipus descrit a la figura 1 (Suñé, 2015)

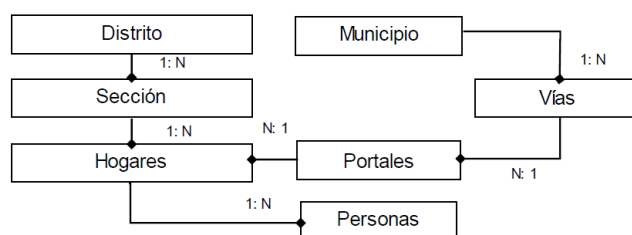


Figura 1 Base de dades associada al Registre d'Habitants (Suñé, 2015)

L'adreça postal permet geocodificar aquesta informació dintre d'un marc de referència espacial oferint la possibilitat d'analitzar, consultar i interpretar aquesta informació geogràficament.

Aquest treball s'ha elaborat a partir del Registre de la Població geocodificat de l'any 2014.

1.3.-Representació espacial del Registre de la Població

Històricament, el tractament de les adreces postals conduïa a l'assignació de zones administratives o zones censals. Això comportava que els resultats estadístics només es podien comptabilitzar en aquestes zones, fet que donava lloc a una escala de treball força limitada i una pèrdua d'informació espacial força alta. Davant d'aquesta situació, es va apostar per la geocodificació de les adreces postals. (Suñé, 2016)

La geocodificació de les adreces dels portals va iniciar una nova problemàtica, perquè la informació resultant (de gran resolució espacial) és massa sensible al secret estadístic. Per tant, no es poden fer públiques directament. Cal trobar-hi una solució on hi hagi un equilibri entre el secret estadístic i la resolució de la informació espacial.

Després de diferents estudis del grup de recerca *Data Analysis and Modeling Research Group de la Universitat de Vic* (Lagonigro Raymond et al, 2016), es van trobar dues solucions metodològiques possibles: la pertorbació de coordenades i l'agregació espacial (Suñé, 2016).

Idescat va concloure que la metodologia més adequada és l'agregació espacial variable (passar la informació de la capa de punts d'adreces de la població a una capa de polígons que contingués la informació agregada). Aquesta metodologia utilitza un sistema de multiresolució espacial basat en quadrees.

1.4.-Mètodes de representació espacial

Tot i que Idescat va seleccionar el mètode de representació basat en quadrees, en aquest treball s'ha representat cartogràficament el Registre de la Població amb els dos mètodes: per una banda els quadrees i per altra banda la pertorbació de coordenades.

1.4.1.-Representació mitjançant quadrees

El quadtree és una estructura jeràrquica en què cada geometria (pare), un quadrat, es divideix en quatre geometries més petites (fills), quatre quarts. La divisió d'aquesta estructura jeràrquica s'adapta de forma local per preservar el secret estadístic a partir d'un llindar de població. Així doncs, si hi ha suficient població en una geometria es divideix l'àrea d'aquesta en quatre fills. El procés es fa recursivament sempre que totes les geometries continguin una població major que el llindar establert.

Aquesta representació transforma les unitats puntuals (Registre de la Població) en unitats geomètriques poligonals (elements del quadtree).

El quadtree va ser la representació que va seleccionar Idescat per representar espacialment el Registre de la Població. Idescat va crear l'any 2016 un quadtree a partir del Registre de la Població de l'any 2014. Aquest quadtree va ser creat amb diferents paràmetres, però sense poder analitzar si eren els més òptims o no.

L'objectiu del present treball és descobrir quin és el valor òptim de cadascun dels paràmetres de creació dels quadrees per tal de construir el quadtree més òptim.

1.4.2.-Representació mitjançant la pertorbació de coordenades

La pertorbació de coordenades és el mètode pel qual es modifiquen les coordenades de la capa de punts. Aquesta pertorbació es pot implementar de moltes formes diferents, però en aquest treball s'ha fet una pertorbació de coordenades amb la mínima pertorbació possible, intercanviant les coordenades per parelles de punts amb atributs semblants i a més, que siguin pròxims entre ells.

La pertorbació de coordenades és un altre mètode per a representar espacialment el Registre de Població sense vulnerar el secret estadístic.

Abans d'aquest treball, Idescat no havia realitzat la representació mitjançant la pertorbació de coordenades, per tant, aquest treball serveix com una prova o aproximació a aquesta representació.

2.-Objectius del treball

2.1.-Objectius

El present treball té dos objectius generals. El primer és la determinació dels paràmetres més òptims per a la creació d'un quadtree a partir de la informació de la població del RP2014 geocodificada a través de l'anàlisi dels errors dels resultats per mitjà del mètode de Montecarlo. Per a assolir aquest objectiu general, cal complir els següents objectius específics:

- Obtenció de la geometria del quadtree i de la relació recursiva de les geometries pares i fills .
- Obtenció del paràmetre del llinar de població utilitzant la geometria del quadtree i la informació del RP2014 geocodificada.
- Obtenció dels paràmetres de la resolució màxima i mínima en funció dels resultats obtinguts en el primer objectiu específic.
- Creació de diferents quadrees amb diferents llinars i resolucions.
- Obtenció de l'error relatiu dels quadrees a partir del mètode de Montecarlo.
- Anàlisi dels errors i determinar quins són els paràmetres més òptims per a la creació del quadtree.

El segon objectiu és l'assaig de la pertorbació de coordenades per a la representació espacial de les dades del Registre de la Població. Per assolir aquest objectiu general, cal resoldre els següents objectius específics:

- Creació del dos conjunt de dades on s'inclogui la informació resumida del Registre de Població (*arrays*)
- Creació del codi Java per trobar el parell de coordenades per cada unitat puntual a partir d'atributs semblants i per proximitat espacial.
- Anàlisi de l'error dels atributs de pertorbació de coordenades en relació als atributs de les dades sense pertorbar a partir del mètode de Montecarlo.

2.2.-Abast

L'estada a Idescat ha estat des del 18 d'abril fins el dia 16 de Juny. Des del inici de les pràctiques fins el dia 17 de maig s'ha realitzat la creació i l'anàlisi d'errors dels quadrees. A partir del 18 de maig s'ha realitzat l'assaig i l'anàlisi d'errors de la pertorbació de coordenades.

2.3.-Pla de treball

En aquest apartat s'adjunta quin ha estat el pla de treball segons els diferents objectius citats anteriorment.

Objectius	Data inici	Data final	Duració (días)
Obtenir geometria	18/04/2017	20/04/2017	2
Obtenir llinar	20/04/2017	24/04/2017	4
Obtenir resolució màxima	20/04/2017	24/04/2017	4
Obtenir resolució mínima	20/04/2017	24/04/2017	4
Creació de les funcions per elaborar el QT	24/04/2017	08/05/2017	14
Creació dels quadrees	08/05/2017	10/05/2017	2
Anàlisi errors dels quadrees	10/05/2017	17/05/2017	7
Creació del dos conjunt de dades (arrays) del RP	18/05/2017	20/05/2017	2
Creació del codi Java per trobar el parell de coordenades	21/05/2017	10/06/2017	20
Anàlisi de l'error dels atributs de pertorbació	11/06/2017	16/06/2017	5

Figura 2 Pla de treball

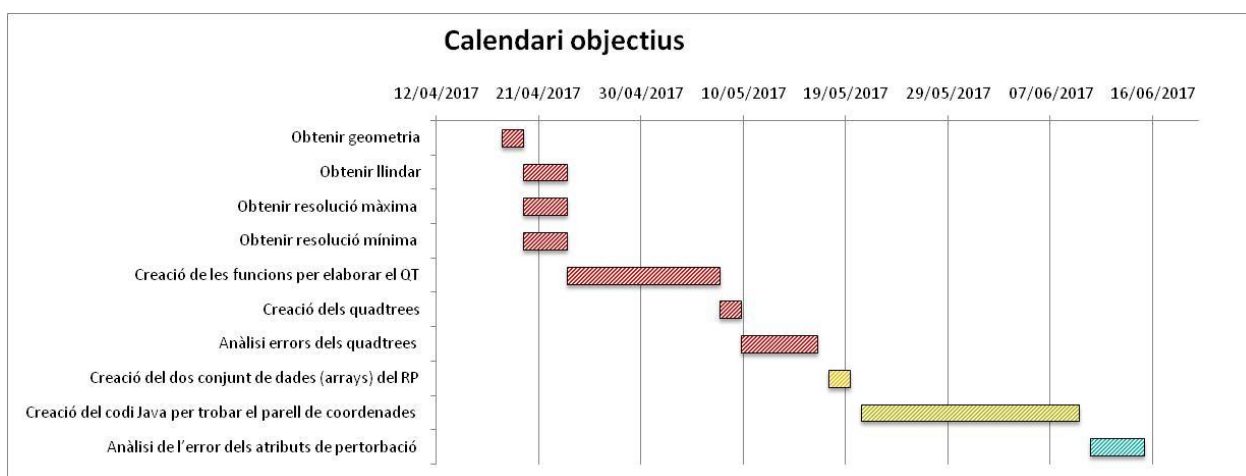


Figura 3 Calendari dels objectius

3.-Marc teòric

En aquest apartat s'exposen els principals conceptes sobre els quadrees, el mètode de Montecarlo i la pertorbació de coordenades.

3.1.-Quadrees

Els quadrees, o arbres quaternaris, són estructures de dades espacials d'arbres jeràrquics que es basen en el principi de la descomposició recursiva de l'espai. La divisió d'una regió crea quatre regions (fills) que corresponen a sud-oest, nord-oest, sud-est i nord-est. (Aluru, 2005).

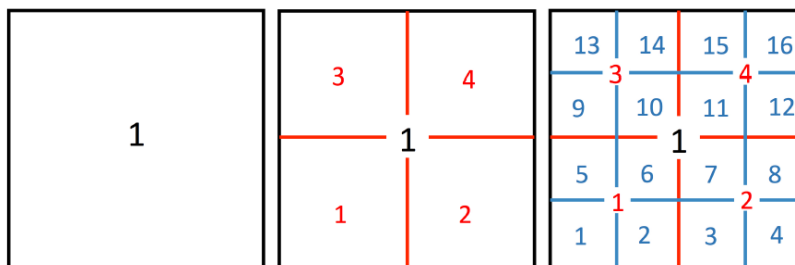


Figura 4 Exemple de divisió recursiva de l'espai (Lagonigro Raymond et al, 2016)

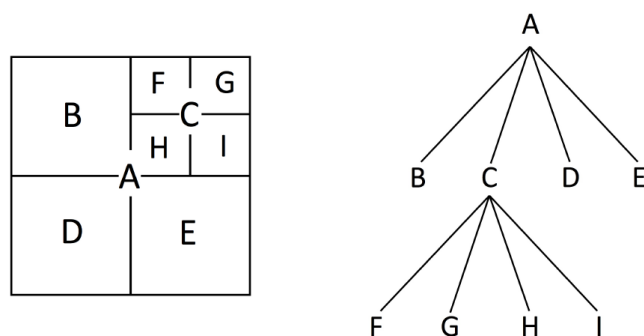


Figura 5 Exemple de divisió recursiva de l'espai II (Lagonigro Raymond et al, 2016)

El terme quadtree va ser proposat per Raphael Finkel i J.L. Bentley l'any 1974, que van crear aquesta estructura de dades per a emmagatzemar informació i recuperar-la de forma fàcil i ràpida gràcies a la clau composta única que correspon a cada element. (R. A. Finkel et al, 1974).

A partir de l'anterior estudi, van sortir molts estudis referents als diferents mètodes de creació i els diferents algorismes de treball amb quadrees. En tots aquests estudis es va perseguir la creació de diferents estructures dinàmiques multidimensionals (Mark de Berg et al, 2008). En

la majoria de casos, el quadtree tenia l'objectiu d'indexar informació per més tard, ser consultada (David Eppstein et al, 2005).

Però aquesta estructura d'informació va adoptar un nou significat quan altres disciplines van importar aquesta metodologia al seu camp.

En el cas de la informació geogràfica i de la representació cartogràfica, es va veure que els quadtrees oferien la possibilitat de resumir la informació del territori en una malla multiresolució. Seguint aquesta línia, es van començar a fer estudis per resumir la informació demogràfica (Martin Behnisch et al, 2013). Aquest resum es produeix a partir de transformar la informació puntual a atributs dels elements del quadtree a partir de la intersecció d'aquests punts amb la geometria del quadtree i del recompte del nombre de punts o del sumatori dels atributs dels punts en cada quadrat del quadtree.

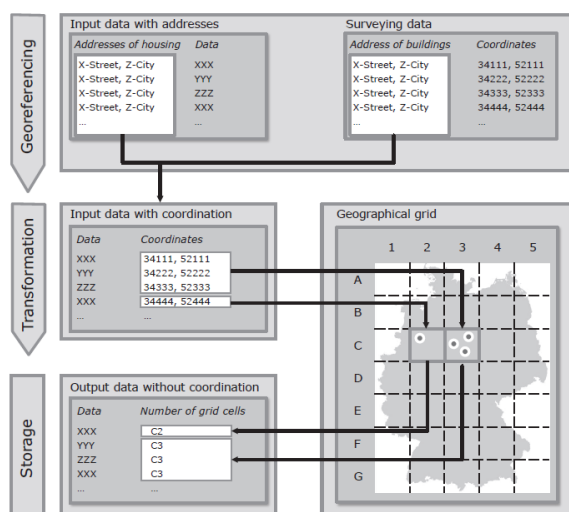


Figura 6 Procés de transformació de dades demogràfiques a quadtree (Martin Behnisch et al, 2013)

En el resum de la informació demogràfica, ja no es fa servir el quadtree únicament com un índex, sinó com un contenidor d'informació. Aquests contenidors, es van agregant amb els seus germans entre si depenent del llinar a assolir de la variable demogràfica representada. (Lagonigro Raymond et al, 2016), (Suñé, 2016).

Aquesta nou objectiu del quadtree ha aportat noves línies d'investigació i de representació de les dades i és dintre d'aquestes línies on s'inclou el present treball.

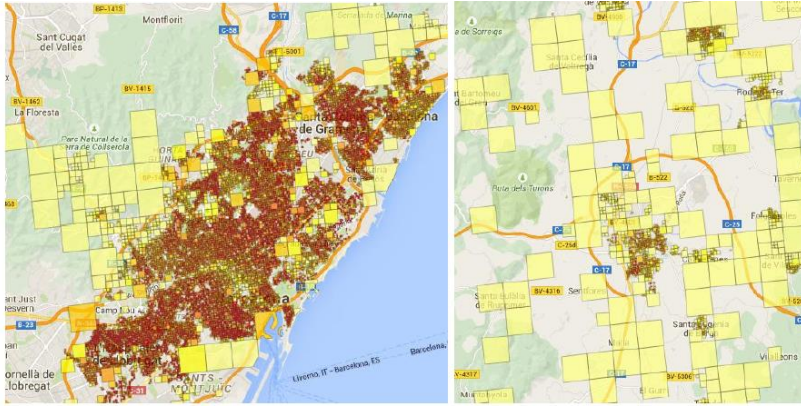


Figura 7 Exemple de quadtree que conté informació demogràfica (Lagonigro Raymond et al, 2016)

Aquesta nova representació geogràfica comporta molts avantatges per a la localització d'informació demogràfica (major resolució i precisió) però també comporta problemàtiques com la preservació del secret estadístic i la metodologia per trobar un llindar i una resolució òptimes (Martin Behnisch et al, 2013), (Suñé, 2016). Idescat és un dels organismes pioners en l'estudi dels avantatges i problemàtiques d'aquesta nova representació, ja que altres estudis d'àmbit demogràfic europeu (Lagonigro Raymond et al, 2016) no aprofundeixen sobre una metodologia clara per a obtenir els paràmetres més òptims.

3.1.1-Paràmetres dels quadrees

Els paràmetres que defineixen un quadtree són la resolució màxima, la resolució mínima, la capa d'elements que cal transformar (o originalment indexar) i el llindar de la variable a representar, en aquest cas la població.

La resolució màxima és la resolució més gran. És a dir, la mida del quadrat més petit. A partir de l'agrupació d'aquests elements s'edifica la resta del quadtree.

La resolució mínima és la resolució més petita. És a dir, la mida del quadrat més gran. És a partir de la divisió d'aquests que es crea, alternativament, la resta del quadtree.

La capa d'elements a transformar són el conjunt d'elements que cal agrupar. En aquest estudi és la capa de punts del Registre de la Població de l'any 2014 geocodificat.

El llindar de la variable a representar és un número per sota del qual cal agregar la informació de la geometria a una geometria d'un nivell superior del quadtree.

3.1.1.1.- Llindar de població

El llindar de la població és el paràmetre que determina si es produeix una agregació. Si dintre d'una geometria, el valor de la població és inferior al llindar, la població serà agregada en els elements geomètrics a una resolució inferior (nivell superior en l'arbre del quadtree).

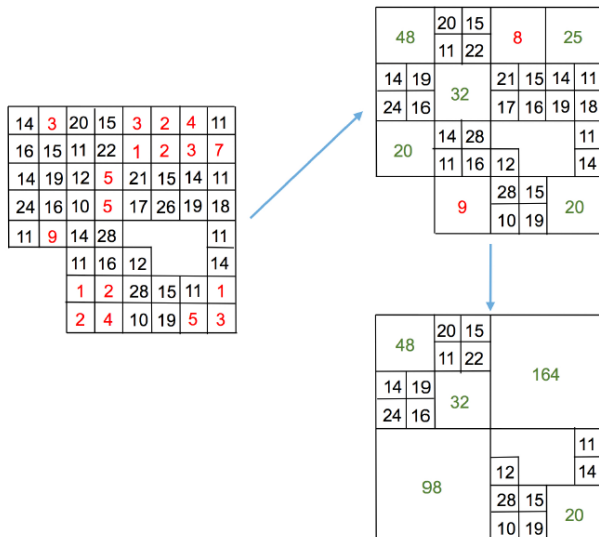


Figura 8 Creació del quadtree a partir de l'agregació espacial amb un llindar de 10 (Lagonigro Raymond et al, 2016)

Com es pot veure a la figura anterior, el llindar determina si es produeix una agregació.

La geometria del primer quadtree és la resolució màxima, que es el nivell més baix de resolució. En aquest nivell veiem que hi ha geometries que inclouen valors de color negre i de color vermell. Les que inclouen valors en vermell vol dir que dintre d'aquesta geometria hi ha menys de 10 elements puntuals, en canvi les que hi ha valors en negre vol dir que hi ha més de 10 elements puntuals.

Com que el llindar és de 10, cal agrupar les geometries que tinguin nombres en vermell amb els seus germans (agrupació) per tal que la suma dels valors que conté cada geometria sigui més gran de 10. Aquest pas es materialitza amb el segon quadtree.

Com que encara hi ha geometries que inclouen valors inferiors a 10, es torna a produir una altra agregació a un altre nivell. El resultat és el tercer quadtree.

En aquest tercer quadtree es pot veure que hi ha geometries de tres nivells diferents, però ara tots els valors de les geometries tenen un valor per sobre de 10.

En aquest cas, en l'última agregació, totes les geometries inclouen més de 10 elements, però no sempre és així. Es pot donar el cas que per culpa de la resolució mínima (geometria en què els quadrats són més grans) encara hi hagin casos on el valor sigui inferior al llindar. En aquests casos, cal amagar aquets valors i posar-los com a nuls per tal que es mantingui el secret estadístic.

Tal i com s'ha pogut veure en l'anterior figura, el llindar és el paràmetre més important a l'hora de crear un quadtree, ja que és el discriminador de l'agregació.

3.1.1.2-Resolució màxima

Com es pot apreciar en la figura 9, la creació del quadtree comença amb el recompte d'elements de cada geometria per al nivell de geometria de resolució màxima (la geometria de resolució màxima és aquella geometria en que els quadrats són més petits).

14	3	20	15	3	2	4	11
16	15	11	22	1	2	3	7
14	19	12	5	21	15	14	11
24	16	10	5	17	26	19	18
11	9	14	28				11
		11	16	12			14
		1	2	28	15	11	1
		2	4	10	19	5	3

Figura 9 Detall de la creació del quadtree a partir de l'agregació espacial amb un llinar de 10 Font: (Lagonigro Raymond et al, 2016)

La resolució màxima és el paràmetre inicial, ja que és la base sobre la qual s'inicia el procés d'agregació. Si la resolució màxima és petita, es a dir, les dimensions dels quadrats mínimes és gran es perd precisió. En canvi si és gran (la dimensió dels quadrats mínims és petita) s'augmenta el risc de revelació.

3.1.1.3-Resolució mínima

La resolució mínima és la resolució en què les geometries del quadtree són més grans. Aquesta és la geometria inicial i és a partir de la divisió d'aquesta geometria que comença el procés de creació del quadtree.

Es pot arribar a pensar que com més gran siguin aquest polígons de la geometria, més gran serà la recursivitat del quadtree, però cal pensar també que els casos on la població és inferior al llinar en la resolució mínima, fan augmentar l'error quan un usuari cavi càlculs de la població en les seves zones d'interés.

Per tant, cal decidir mitjançant els diferents quadrees creats quina és la resolució mínima més adient.

3.2.-Mètode de Montecarlo per a l'estimació de l'error.

Un cop aplicats els quadrees i la pertorbació de coordenades, cal fer una estimació de l'error en totes dues metodologies.

En el cas dels quadrees, el problema de l'estimació del error apareix quan un usuari vol operar amb la informació del quadtree. Partim de la premissa que l'usuari no vol fer una consulta per

una unitat de geometria del quadtree, sinó que ho voldrà fer per una àrea del seu interès. Aquestes àrees d'interès, poden intersecar de múltiples formes amb el quadtree

A més els habitants que hi ha dintre de la geometria no tenen per què estar distribuïts de manera uniforme en tota la geometria. Per tant, al calcular la població dins l'àrea d'interès del usuari, es farà la suposició de que la població es distribueix uniformement en la superfície dels elements del quadtree i el resultat tindrà un cert error.

Com que aquest error depèn principalment de les característiques de forma de les àrees d'interès, s'ha utilitzat la metodologia de Montecarlo per obtenir les àrees d'interès.

En el cas de la pertorbació de coordenades, l'error no està en el valor de la població, ja que la població no està agregada en unitats superiors, sinó que continua sent una capa de punts. L'error està en els atributs de les unitats puntuals resultants de la pertorbació respecte a les unitats puntuals originals.

Per estimar l'error de la pertorbació de coordenades també s'ha utilitzat la metodologia de Montecarlo per obtenir les àrees d'interès.

El model de Montecarlo permet estimar el comportament d'una variable, en aquest cas la forma de les àrees d'interès i l'error derivat de la intersecció amb la geometria del quadtree o l'error derivat dels atributs amb el canvi de coordenades. El procediment d'aquest mètode és crear diferents escenaris possibles i veure com es comporten. (Agüí, 2008)

El mètode de Montecarlo té el nom degut al Casino de Monte Carlo i en concret a la ruleta, ja que aquest joc es basa en nombres obtinguts de forma aleatòria. El mètode de Montecarlo es pot aplicar a molts estudis diferents, el més famós va ser l'estudi per la creació de la bomba atòmica, ja que va simular els problemes de difusió dels neutrons.

Per saber l'error derivat de les dues metodologies, s'han generat un total de 50.000 polígons aleatoris per simular les àrees d'interès que poden arribar a dibuixar els usuaris.

3.3.-Pertorbació de coordenades

La pertorbació de coordenades és un mètode força complex que està relacionat amb els valors dels atributs, la distribució espacial de les dades i el nivell de pertorbació que volem aplicar.

L'objectiu de la pertorbació de coordenades és que no es sigui capaç de tornar a l'estat o posició inicial de les dades, ja que sinó és podria comprometre la confidencialitat d'aquestes (Young, 2009). Però també cal trobar un equilibri, ja que si es pertorba massa el resultat final pot ser fictici o irreal comparat amb l'estat original.

Han estat molts els autors que han desenvolupat diferents algorismes i teories per trobar una pertorbació de coordenades adequada, i entre aquests hi ha massa diferències metodològiques, ja que depenen de la naturalesa de les dades i de les restriccions legals de cada país, s'han de seguir unes directrius diferents. (Lagonigro Raymond et al, 2016)

En el cas de la informació geogràfica, la pertorbació de coordenades pot ser un inconvenient, ja que si canviem les coordenades, pot ser que les noves coordenades caiguin sobre llocs inhabitats o també en accidents geogràfics com rius o mars. (Lagonigro Raymond et al, 2016)

Per tant, cal establir que la pertorbació de coordenades es produeix mitjançant el canvi dels atributs entre punts pròxims entre si. Així es preserva el secret estadístic i s'evita el canvi de coordenades massa allunyat. Aquest canvi d'atributs pot crear noves problemàtiques perquè aquest canvi pot no produir-se degut a la incompatibilitat o al baix nivell d'aparellament de punts amb atributs similars.

Aquesta pertorbació de coordenades on es canvien els atributs dels punts pels atributs de punts pròxims a ells s'anomena *switch* de coordenades o intercanvi de coordenades.

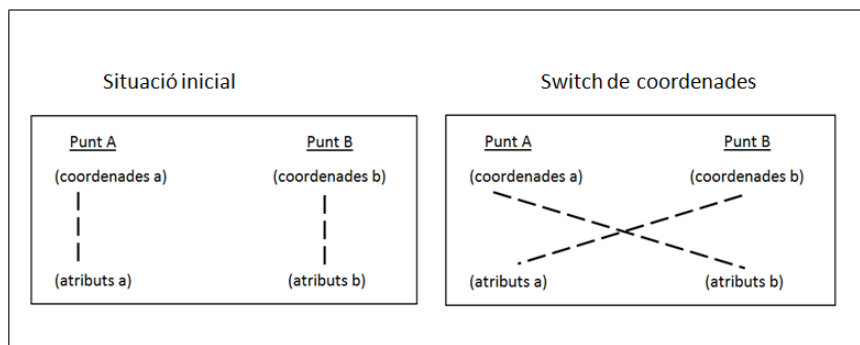


Figura 10 Esquema d'intercanvi (*switch*) de coordenades

En l'anterior figura es pot veure com les coordenades dels punts no canvien, sinó que són els atributs els que canvien entre un parell de punts, però de fet el resultat és equivalent a un intercanvi de coordenades ja que és com si l'individu A es desplaçés a la posició de l'individu B i viceversa. Aquesta metodologia d'intercanvi(*switch*) de coordenades és la que s'ha utilitzat en aquest treball.

4.-Dades d'Idescat

Idescat ha facilitat tota la informació necessària per a aplicar els quadrees i la pertorbació de coordenades i fer el càlcul dels errors de les dues metodologies.

Entre aquestes dades hi ha el Registre de la Població geocodificat del 2014, les geometries de malla de resolucions diferents, que serviran per a generar els quadrees mitjançant agrupació, una taula que relaciona les geometries entre si i els polígons aleatoris generats amb el mètode de Montecarlo.

4.1.-Registre de la Població

El Registre de la Població seleccionat per fer aquest treball és el Registre de la Població de l'any 2014. Aquest Registre de la Població es va geocodificar per Idescat l'any 2015. En total hi ha 7.566.464 registres, corresponents a habitants individuals.

Aquest registre està materialitzat en una taula amb el nom HABITANTS_EDAT_POSICIO.

4.2.-Geometria dels quadrees

Idescat ha facilitat les geometries que va utilitzar per a realitzar el seu estudi previ l'any 2016. En concret ha facilitat la geometria de les malles de quadrats a una resolució de 125, 250, 500 i 1000 metres en l'àmbit de Catalunya.

La geometria de malla de 1000 metres correspon al nivell 0, la geometria de malla de 500 metres al nivell 1, la geometria de malla de 250 metres al nivell 2 i la geometria de malla de 125 metres al nivell 3.

Aquestes geometries van ser creades a partir de la divisió recursiva de la geometria "EuropeanWorkshop on ReferenceGrids in 2003" de 1000 metres definida per EuropeanEnvironmentAgency (EEA).

Les geometries estan incloses dintre de la taula GRID_ETRS89_LALEA_MULTIRESOL i estan en el sistema de coordenades projectades ETRS89_LAEA (Lambert azimuthalequal-areaprojection,EPGS:3035).

La relació recursiva que identifica les geometries inicials (pares) amb les seves 4 divisions successives (filles) estan resumides en una taula. Aquesta taula és la materialització d'aquesta recursivitat 1:N, on per a cada fill s'indica quin és el seu pare. Aquesta taula s'anomena GRID_PARES_FILLS.

4.3.-Polígons de Montecarlo

Idescat ha facilitat els 50.000 polígons creats a partir del mètode de Montecarlo. El fet que els polígons siguin els mateixos que es van utilitzar per a calcular els errors dels seus quadrees, fa que els resultats que s'han obtingut d'aquest treball siguin comparables amb els seus.

5.-Metodologia

En aquest apartat s'exposa la metodologia per a la creació i validació dels quadrees i també per a la creació i validació de la pertorbació de coordenades.

5.1.-Metodologia de creació i validació dels quadrees

En aquest apartat hi ha l'explicació en detall sobre la metodologia de creació i validació dels quadrees. Aquesta explicació comprèn el mètode de construcció dels quadrees, les variables necessàries per a la construcció del quadree i l'anàlisi dels errors del resultat.

5.1.1.-Creació dels quadrees

La creació dels quadrees s'ha fet a partir de tres passos. El primer pas ha estat calcular el nombre d'habitants que hi ha dintre dels quadrats de la resolució màxima.

El segon pas ha estat a partir de la relació recursiva entre les geometries pares i les geometries fills, calcular el nombre d'habitants en totes les altres resolucions.

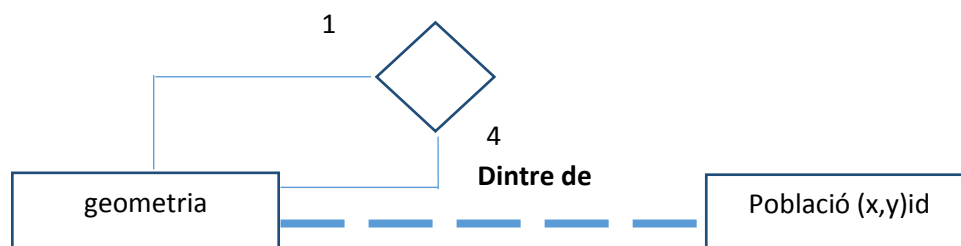
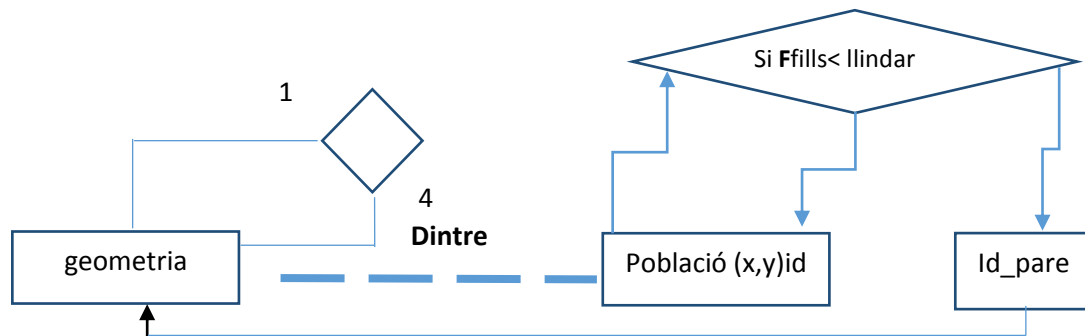


Figura 11 Diagrama de creació dels quadrees

L'anterior figura és un diagrama que mostra els dos primers passos. Amb aquests dos primers passos es pot saber quina és la població agregada que hi ha dintre de cada quadrat de la malla a les diferents resolucions.

El tercer pas ha estat avaluar la població de cada quadrat de cada malla començant per la resolució mínima. Si la població està per sobre del llindar, indica que es pot dividir, per tant s'avaluen els seus 4 fills. El procés s'atura quan a l'avaluar una geometria pare, un dels seus fills està per sota del llindar.



En l'anterior diagrama es veu com la creació del quadtree comença per avaluar la població que hi ha dintre de la geometria. Per qüestions tècniques, es comença avaluant els quadrats amb una resolució mínima (quadrats més grans). Si la població agregada d'aquests quadrats està per sobre del llindar, s'avaluen els quatre fills, després, si es continua complint la condició que la població és superior al llindar, s'avaluen els quatre fills del primer fill, després els del segon, etc.

Lògicament, aquest procés s'efectua per a cada combinació dels paràmetres que determinen el quadtree (lindar de població, resolució màxima i resolució mínima), de manera que es creen tants quadrees com combinacions de paràmetres es decideix assajar.

5.1.3.-Avaluació de l'error dels quadrees

Un cop creats els diferents quadrees, cal efectuar un seguit de proves per estimar l'error que obtindríem si calculéssim la població d'una àrea aleatòria sobre cada un dels quadrees creats amb diferents paràmetres de llindar de població i resolució màxima. L'estimació d'aquest error i la posterior comparació d'aquest error entre els diferents quadrees (creats amb altres paràmetres) ens permet saber quins són els valors dels paràmetres del quadree que donen lloc a uns errors més baixos.

Cada quadtree conté la deslocalització de la geocodificació de la població. Aquesta deslocalització condueix a assumir que la població que hi ha dintre de la geometria del quadtree està repartida uniformement per tota la geometria. Aquest fet és erroni, perquè pot ser que la població no estigui repartida de manera uniforme. Per això a l'intersecar la informació del quadtree amb una àrea aleatòria pot comportar errors en el càlcul de la població d'aquesta àrea.

El fet de transformar una capa de punts en una capa de polígons mitjançant l'agregació de les dades dels punts, pot provocar errors en l'obtenció d'informació en una àrea d'interès si aquesta àrea no coincideix exactament amb les geometries del quadtree, cosa pràcticament impossible per a àrees irregulars.

Depenent de l'àrea d'interès i dels paràmetres que defineixen el quadtree, els errors poden ser molt diferents entre els quadrees creats. Per poder estimar aquest error cal crear aleatòriament diferents àrees d'interès i interseccar-les amb els quadrees creats anteriorment i

a partir de la comparació d'aquests errors mitjançant diferents tractaments estadístics , podrem determinar quins són els paràmetres més idonis per a la construcció del quadtree òptim.

Per a simular les àrees d'interès s'han construït un total de 50.000 polígons. Aquests polígons van ser creats per Idescat en estudis anteriors i s'han utilitzat ara per poder comparar els resultats obtinguts amb els resultats dels estudis realitzats per Idescat anteriorment.

Aquests polígons han estat creats a partir de la generació d'un centre x,y aleatori, la selecció aleatòria d'un radi i una divisió en n aleatori sectors, la selecció aleatòria d'un punt en cada sector i finalment la unió dels sectors corresponents als punts seleccionats.

5.2.-Metodologia de la pertorbació de coordenades i la seva validació

En aquest apartat hi ha l'explicació en detall sobre la metodologia d'aplicació de la pertorbació de coordenades i de la seva validació

5.2.1.- Creació de l'espai de publicació i l'espai de comparació

El registre de població, com ja s'ha comentat conté les variables sexe, edat, nacionalitat i lloc de naixement a més de l'adreça que dona lloc a les coordenades quan es fa la geocodificació. Per tant, si el criteri d'intercanvi de posicions es fa entre persones que tinguin els mateixos valors de les variables menys una, qualsevol resultat que no inclogui aquesta variable donarà un resultat amb error zero.

D'aquestes variables, el candidat a priori més adequat és l'edat ja que es tracta d'una variable continua i permetria afinar a voluntat la magnitud de la pertorbació.

En el conjunt de variables originals la difusió de dades ha de ser tal que les freqüències mínimes de casos diferents han de ser superiors a un cert valor (k-anonymity). Això portat al nostre conjunt de dades fa que les variables hagin d'estar codificades en els valors:

Sexe: home, dona

Edat: quinquennal

Lloc de naixement: Catalunya, resta de l'estat, estranger

Nacionalitat: espanyol, estranger

A aquest espai de mesura (E) el denominarem espai de publicació ja que ens permetria, si no fos perquè conté les posicions, la seva difusió preservant el secret estadístic.

Suposem doncs que definim un altre espai, proper a E, que denominarem espai de comparació (E'), consistent en la recodificació de les edats mantenint la resta d'atributs sense variar:

Sexe: home, dona

Edat: desenal

Lloc de naixement: Catalunya, resta del estat, estranger

Nacionalitat: espanyol, estranger

5.2.2.-Funció d'intercanvi de coordenades

L'intercanvi de coordenades entre dos punts a i b s'ha fet mitjançant la funció:

$$P(a,b) \begin{cases} = 0 & \text{si } E(a) = E(b) \\ = 0 & \text{si } E'(a) <> E'(b) \text{ i } E(a) <> E(b) \\ = f(1/d(a,b)) & \text{si } E'(a)=E'(b) \text{ i } E(a) <> E(b) \text{ i } d(a,b) > 0 \\ = 0 & \text{si } E'(a)=E'(b) \text{ i } E(a) <> E(b) \text{ i } d(a,b) = 0 \end{cases}$$

Figura 13 Funció d'intercanvi de coordenades entre els punts a i b

És a dir, l'intercanvi de les coordenades és possible entre punts iguals en l'espai de comparació però diferents en l'espai de publicació sempre que la seva distancia ($d(a,b)$) sigui més gran que zero.

La funció $F(1/d(a,b))$ ens permet escollir aleatòriament un dels possibles candidats fent que sigui més probable l'intercanvi entre punts més propers.

Cal tenir en compte que el nombre de punts del conjunt de dades (uns set milions) fa que el càlcul no sigui viable si no limitem el nombre de candidats en l'intercanvi. El límit escollit ha estat de 25 punts. És a dir, per a cada punt hem cercat els 25 punts més propers que complien les condicions descrites anteriorment (igualtat en l'espai de comparació però no en el de publicació). De tal forma que el intercanvi es produeix al seleccionar un dels 25 elements de forma aleatòria fent que la probabilitat d'assignació sigui funció del invers de la distància.

Tot i això el càlcul és extremadament llarg i no hem tingut temps material per obtenir un resultat del conjunt del total de punts. Mostrarem per tant els resultats d'una part que creiem significativa del total (2.382.186 punts).

5.2.3.- Avaluació de l'error de la pertorbació de coordenades

Per mesurar els efectes de la pertorbació, hem utilitzat el conjunt de 50.000 polígons aleatoris utilitzats en els experiments de Montecarlo anteriors. Sobre cada un d'aquest polígons s'ha calculat la distribució de les edats, tant per a la capa original com per a la pertorbada.

Per mesurar la semblança entre la capa original i la capa amb els efectes de la pertorbació hem utilitzat la distància de khi quadrat. Aquesta distància s'ha calculat sobre els vectors de freqüències de classes normalitzats de les dues capes.

6.- Preparació de les dades i obtenció dels paràmetres previs a la creació dels quadrees

Cada quadtree es defineix segons els paràmetres de resolució màxima, resolució mínima, els objectes puntuals a agrupar (en aquest cas els registres de la població geocodificats) i el llinar de població.

Abans de crear els quadrees cal esbrinar quins són els valors dels paràmetres que podíem arribar a pensar que són més òptims.

6.1.-Determinació dels llinars més adients

Per identificar el llinar de població més òptim cal primer seleccionar uns quants llinars de població diferents per tal de poder crear diferents quadrees i veure quin dels llinars triats és el més òptim. Per trobar els diferents llinars a comparar, s'ha fet una anàlisi de freqüències de la capa de punts de la població de 2014 amb totes les geometries dels diferents nivells del quadtree. El primer pas ha estat importar aquesta capa de punts, després fer la intersecció de la capa de punts amb les geometries a diferents nivells i per últim comptar les freqüències i analitzar-les.

6.1.1-Creació de la taula del Registre de Població geocodificat

El primer pas ha estat la càrrega de la taula del Registre de Població de la bases de dades d'Idescat a la nostra base de dades de treball de PostgreSQL. Aquesta taula importada l'he anomenat HABITANTS_EDAT_POSICIO_UTM.

El més important d'aquesta taula és la columna POSICIO, ja que aquesta conté la posició georeferenciada dels habitants. Aquesta columna està en el sistema de coordenades ETRS89 UTM 31N. Aquesta columna serà la utilitzada en els següents apartats per fer la intersecció amb la geometria del quadtree.

L'script d'aquest procés és l'Script de SQL 1.

6.1.2.-Intersecció de les dades de la població amb la geometria del quadtree

A continuació es van interseccionar les dades de localització de la població amb les geometries de tots els nivells diferents del quadtree.

L'script d'aquest procés és l'Script de SQL2.

Si es compara la suma del total de la població agregada per a un nivell i la capa de població, es comprova que hi ha un error de 21 persones. Aquest error és produït pel fet que aquestes 21 persones s'han geocodificat fora de les fronteres de Catalunya.

6.1.3.-Anàlisi de freqüències de la població segons la geometria a diferents nivells

Per a l'anàlisi de les freqüències de la població i de les freqüències acumulades de la població agrupades pel nivell de la geometria, s'ha executat el següent script de SQL (Script de SQL 3) En aquest script, es fa una selecció de la població, la freqüència d'aquesta població (*COUNT*) i la freqüència acumulada de la població (*COUNT*POBLACIO*).

La forquilla de població s'ha agafat per sobre de 0 i inferior a 40. La justificació recau en el fet que només volem saber si la freqüència de la població té un punt crític per sobre del qual hi hagi una alta freqüència de la població.

Aquest script s'ha repetit un total de cinc cops, un per a cada nivell de la geometria.

Les anteriors dades s'han representat en un diagrama lineal, en el qual l'eix de les x representa la població i l'eix de les y representa la suma de la freqüència acumulada de la població.

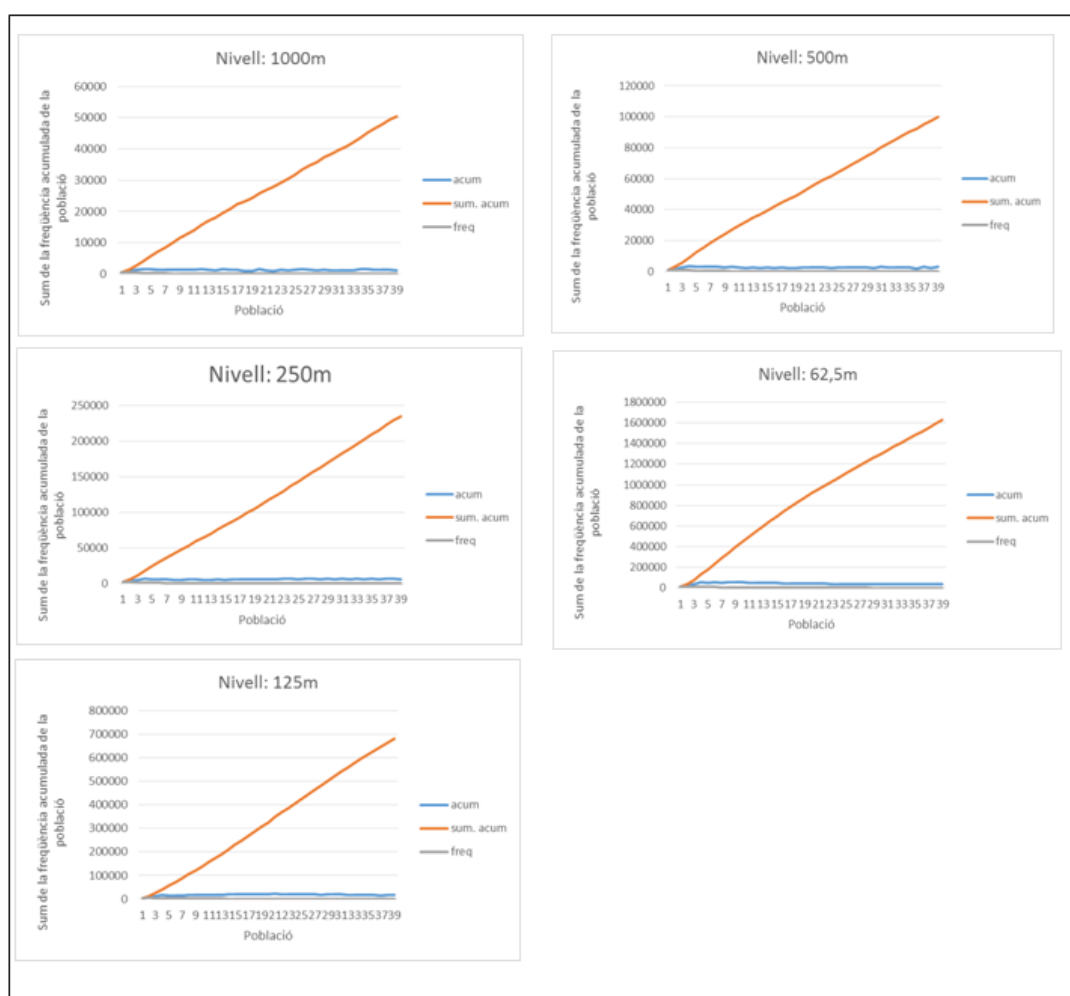


Figura 14 Representació lineal de la suma de la freqüència acumulada de la població

En la figura anterior es pot apreciar que no hi ha cap llindar de la població que sobre el qual hi hagi un canvi de tendència en la corba. En tots els gràfics la corba es completament lineal i no hi ha cap punt d'inflexió que faci escollir un de particular.

Aquesta mateixa problemàtica sobre la tria del llindar està present en molts estudis (Lagonigro Raymond et al, 2016), (Martin Behnisch et al, 2013). En tots aquets estudis, els autors parteixen d'uns llindars aleatoris per a elaborar els quadrees. En els dos articles s'arriba a la mateixa conclusió òbvia, com més baix és el llindar, hi ha un major nombre d'elements geomètrics a la resolució màxima.

Per tant, donat que no hi ha una metodologia establerta per a determinar el llindar de la població més adequat, he seleccionat un total de quatre llindars diferents (13,17,19 i 23). Els he seleccionat perquè en els quadrees elaborats per Idescat es va seleccionar el llindar de població 17, per tant, he seleccionat els nombres primers més pròxims a ell.

6.2.-Determinació de la resolució màxima

En l'apartat 4.2 d'aquest treball, s'ha exposat que Idescat va crear un total de 4 resolucions de malla (1000,500,250 i 125 metres respectivament) en assaigs anteriors. Idescat va decidir que la resolució màxima en aquells assaigs fos la de 125 metres i la resolució mínima la de 250 metres.

Per a aquest treball, es va decidir crear una nova resolució de malla (62.5m) perquè en els estudis realitzats per Idescat a la resolució de malla de 125 metres hi ha agregacions de població molt per sobre de 17, indicant que una resolució de malla inferior (62.5 metres) és factible.

No es va crear una resolució de malla per sota de la de 62.5 metres ja que aquesta tindria una mida de 31.25 metres i es va decidir que hi haurien molt poques zones de Catalunya que en una malla de mida 31.25 metres hi hagués una població agregada per sobre del llindar de població 17.

En els següents apartats, s'explica quin és el procés de creació d'aquesta nova geometria de malla de 62.5 metres.

6.2.1-Creació de nova geometria

Tal com s'ha parlat en la metodologia de la resolució màxima, es va decidir crear una nova geometria de malla amb una mida de costat de 62.5 metres. Aquesta nova geometria s'ha creat a partir de la geometria disponible a Idescat.

Aquesta malla s'ha creat a partir de les "x" mínima i la "y" mínima originals de la geometria d'1 km. Aquest procés hauria estat més fàcil si la divisió s'hagués dut a terme directament a partir de la mateixa geometria en comptes de fer-ho a partir de les coordenades, però com que la geometria està retallada segons el límit de Catalunya, la transformació s'hagués vist compromesa.

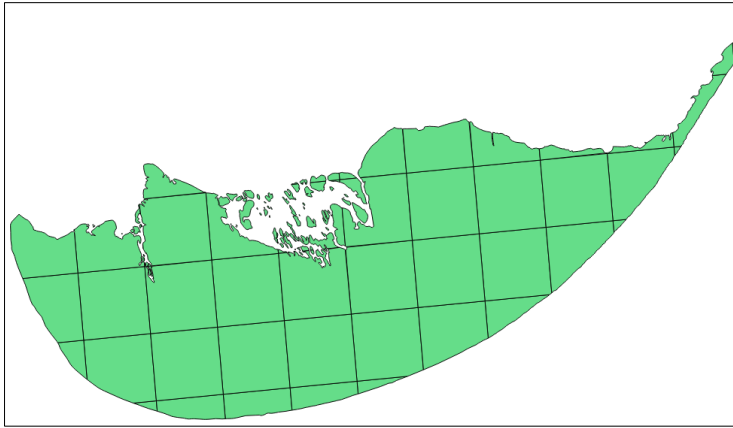


Figura 15 Exemple de la geometria d'1km al Delta de l'Ebre

En aquesta figura podem observar com la geometria d'1 km està retallada a causa de la forma irregular del límit de Catalunya. Per aquest motiu, tal com s'ha explicat anteriorment, s'ha dividit la geometria partint de les coordenades originals de la capa sense retallar i posteriorment s'ha retallat la nova malla pel límit de Catalunya.

6.2.2.-Creació de la taula on s'agrega la nova geometria

Per a generar aquesta geometria, s'ha començat per crear una capa amb els mateixos camps que la capa de la geometria original (GRID_ETRS89_LALEA_MULTIRESOL). Aquesta nova capa serveix per a emmagatzemar la nova geometria creada. Aquesta taula s'anomena TMP_PROVA_DIVISIO.

Aquesta taula té els mateixos camps que la taula GRID_ETRS89_LALEA_MULTIRESOL, però s'ha canviat el camp ID. Aquest camp s'ha convertit d'INTEGER a NEXT VAL, perquè així ens assegurem que l'id s'anirà creant automàticament quan entrem nous registres, en aquest cas, noves unitats de geometria.

6.2.3.-Divisió de la geometria a partir de la funció DIVIDE

Com s'ha descrit anteriorment, la divisió es fa a partir de la geometria d'1km a partir de les coordenades. Per dur a terme aquesta divisió, he creat la següent funció amb el nom DIVIDE, que permet optimitzar el procés de divisió. La funció de SQL de la funció DIVIDE és l'Script de SQL 4.

Aquesta funció demana una total de 9 paràmetres d'entrada. Els més destacats són la x origen, la y origen, la resolució màxima i l'amplada inicial.

El procés de divisió comença quan declarem un seguit de variables (10), de les quals les imprescindibles són: xmin, xmax, ymin, ymax i amplada.

Les xmin i ymin (variables) són la x original i la y original (paràmetres d'entrada), mentre que la xmax i la ymax (variables) són les x i y originals sumant l'amplada (paràmetre d'entrada). L'amplada (variable) serà l'amplada (paràmetre d'entrada) dividit per dos.

Aquest procés s'explica de forma gràfica en les següents figures.

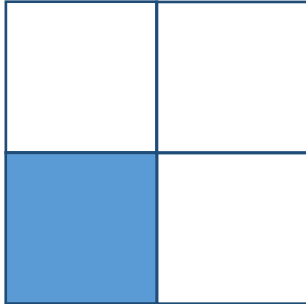


Figura 16 Inici del segon bucle de la funció DIVIDE

A continuació la x es suma l'amplada a la x ($CURRX := CURRX + AMPLADA;$) i és torna a fer aquest segon bucle.

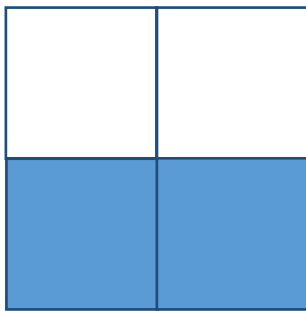


Figura 17 Segon pas del segon bucle de la funció DIVIDE

Si en aquesta divisió l'amplada (variable) és igual a la resolució màxima (paràmetre), voldrà dir que hem generat la geometria de 62.5 metres i per tant, mitjançant una condició(IF) s'afegeix aquesta geometria a la taula creada en l'apartat anterior (TMP_PROVA_DIVISIO.).

En aquest punt hem fet la divisió de la geometria en la x (per tant, tenim dos fills).

Com que la x max, ja es igual que la x, s'acaba el segon bucle i es retorna al primer bucle.

En aquest primer bucle, es suma l'amplada a la y i es torna a iniciar el segon bucle.

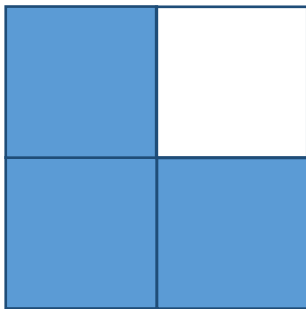


Figura 18 Primer pas del segon bucle de la funció DIVIDE amb la y canviada

Ara es torna a sumar l'amplada ala x i es fa l'últim pas.

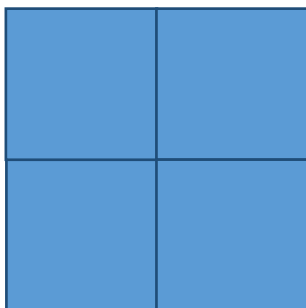


Figura 19 Segon pas del segon bucle de la funció DIVIDE amb la y canviada

D'aquesta manera hem dividit la geometria en els seus 4 fills exactament iguals.

Per poder executar la funció anterior, cal obtenir la xmin i la ymin original. A la taula original GRID_ETRS89_LALEA_MULTIRESOL no hi són, però en la columna GRD_FIXID (identificador del quadrat pare d'una resolució d'1km) hi ha les coordenades originals. Aquestes coordenades, com es pot veure en la figura 20 estan les dues en la mateixa columna, per tant, necessitem dues funcions més, una per a obtenir la coordenada x i una altra per ala coordenada y, que anomenarem GETX_FROM_GRD_FIXID i GETY_FROM_GRD_FIXID (vegeu l'Script de SQL 5 i 6).

	grd_fixid character varying(22)
1	4.0.35990000.21520000
2	4.0.35990000.21520000
3	4.0.35990000.21520000
4	4.0.35990000.21520000
5	4.0.35990000.21520000
6	4.0.35990000.21520000
7	4.0.35990000.21520000
8	4.0.35990000.21520000
9	4.0.35990000.21520000
10	4.0.35990000.21520000

Figura 20 Columna GRD_FIXID de la taula GRID_ETRS89_LALEA_MULTIRESOL, on hi ha les coordenades x i y originals

Un cop tenim les tres funcions, podem executar-les per a obtenir la nova geometria a una resolució de 62.5m.

En aquest punt, s'han d'afegir els paràmetres de la funció *DIVIDE*. Com es pot observar a l'Script de SQL 7, els paràmetres *x* i *y* són les últimes funcions que s'han descrit, la màxima resolució és de 62.5m, i l'amplada és de 1000.

6.2.4-Retall de la geometria segons el contorn de Catalunya

Un cop creada la malla de quadrats de 62.5 metres, s'han retallat aquests per tal que coincideixin amb el límit de Catalunya.

Aquest procés s'ha dut a terme a partir d'una intersecció amb la geometria de la taula *GRID_ETRS89_LALEA_MULTIRESOL*. Aquesta intersecció donarà com a resultat una geometria dels quadrats a una resolució de 62.5 metres retallats pel límit de Catalunya. Aquesta geometria serà guardada dintre de la taula *TMP_PROVA_DIVISIO*, en concret dintre del camp *GEOM*.

Aquesta intersecció s'ha fet amb els quadrats d'1 km (nivell = 0), es podria haver fet amb els de 500 metres o 250 metres, però s'ha trobat més adient utilitzar els d'1km, en tant que originals(vegeu el codi a l'Script de SQL 8).

Un cop feta aquesta intersecció s'eliminen els elements que estan fora del límit de Catalunya. Aquests són els que tenen una àrea (geometria) igual a 0(vegeu el codi al'Script de SQL 9)

6.2.5.-Inserció de la nova geometria a la geometria del quadtree

A continuació s'han inserit aquests quadrats (62.5 metres) a la capa *GRID_ETRS89_LALEA_MULTIRESOL*. Aquest procés s'ha fet a partir d'un *INSERT*, on les columnes de la selecció han de tenir el mateix ordre que les columnes de la segona taula(vegeu l'Script de SQL 10).

Abans d'acabar, fem la unió de la taula *GRID_ETRS89_LALEA_MULTIRESOL* amb la taula *GRID_PARES_FILLS* per poder tenir la relació recursiva del nou nivell de malla afegit(vegeu l'Script de SQL 11).

Posteriorment es comprova que no hi ha cap element geomètric en el nivell 4 que no tingui associat un *id_pare* del nivell 3(vegeu l'Script de SQL 12). Com que en la consulta no surt cap cas en la selecció, això vol dir que no hi ha cap element geomètric en el nivell 4 que no tingui associat un *ID_PARE* de nivell 3

Per facilitar els pròxims passos, com que la geometria de les diferents malles està en el sistema de referència ETRS89:LAEA (Lambert azimuthalequal-areaprojection,EPGS:3035), creem una nova columna (*GEO_UTM*) i transformem la geometria original a la geometria ETRS89_UTM31N (vegeu l'Script de SQL 13). La justificació està en el fet que els punts estan en ETRS89_UTM31N i les geometries en ETRS:LAEA (EPGS:3035).

Abans d'acabar adjunto diferents figures on estan representats en color verd els quadrats de la geometria de malla de 62.5 metres què hi ha una població agregada per sobre del llindar de població 17 i en vermell els que estan per sota.



Figura 21 Vista de Barcelona amb la geometria de malla de 62.5m i la població agregada per sobre del llindar de població 17

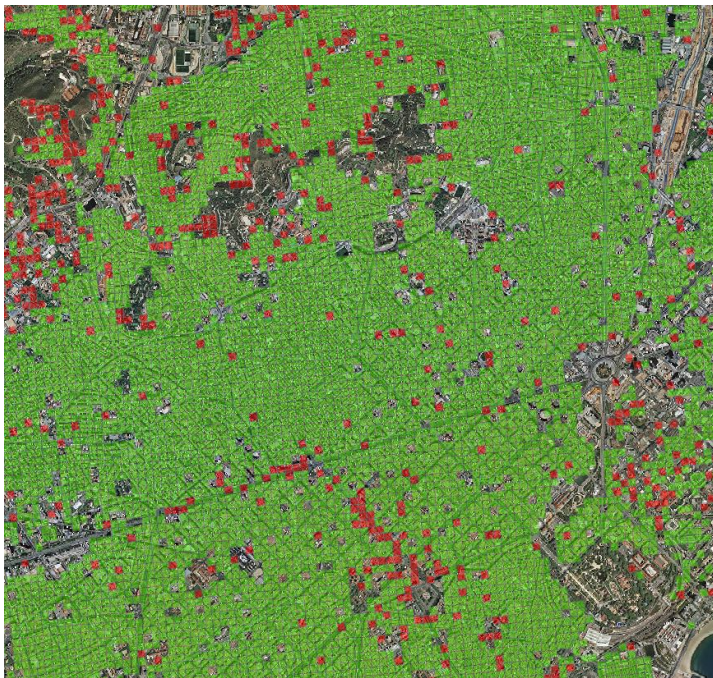


Figura 22 Vista II de Barcelona amb la geometria de malla de 62.5 m i la població agregada per sobre del llindar de població 17



Figura 23 Vista III de Barcelona amb la geometria de malla de 62.5 m i la població agregada per sobre del llindar de població 17

En aquestes figures es pot veure que en el cas de la ciutat de Barcelona, la resolució de 62.5 metres és una resolució màxima òptima, ja que són molt pocs els quadrats que estan per sota del llindar de població 17.

En la figura 23, es pot veure com la resolució espacial de la geometria de la malla de 62.5m és tan alta que es poden diferenciar els patis interiors de les illes de l'Eixample, on no hi ha cap registre de població.

En altres ciutats com Sabadell, Terrassa, Badalona o Mataró, també es pot veure com la resolució de 62.5 metres és una resolució màxima òptima, ja que són molt pocs els quadrats que estan per sota del llindar de població 17.

6.3.- Determinació de la resolució mínima

La resolució mínima que va seleccionar Idescat per a l'elaboració del quadtree en els assaigs de 2016 va ser la de 250 metres. Per a aquest estudi, he trobat adient que la resolució mínima fos la geometria de la malla de 500.

7.-Implementació dels mètodes de quadrees i de la pertorbació de coordenades

7.1.-Implementació dels quadrees

La implementació dels quadrees és el procés de creació de cada un dels quadrees, un cop creada la geometria de la nova resolució màxima i decidits els lindars de població. Aquest apartat s'acaba amb un petit resum dels diferents quadrees creats.

Per a la creació de cada quadree s'ha començat per calcular la informació agregada de la població per als diferents nivells de la geometria. Després a partir d'un llinar, de la resolució màxima i de la resolució mínima se seleccionen els diferents elements (quadrats) que formen el quadree segons aquells paràmetres. Per últim s'unifica en una sola taula, l'emmagatzematge dels quadrats amb la seva geometria i la població agregada dins seu.

Els procediments s'han implementat mitjançant funcions. Això permet que la creació del quadree sigui molt més ràpida i òptima, ja que es crearà de forma automàtica només especificant una geometria, la relació recursiva i els diferents paràmetres del quadree.

Les funcions també permeten que si en un altre moment futur es vol canviar algun paràmetre o valor d'entrada, la seva modificació sigui més ràpida i còmoda.

7.1.1.-Procediment de creació de la informació agregada de la població

A l'executar la funció CREA_TAULA_AGREGATS es crea una taula on hi ha el recompte agregat dels totals dels elements (en aquest cas habitants) i el ID de totes les geometries de tots els nivells.

El primer pas és crear la taula de la informació agregada de la població, el segon pas és omplir aquesta taula amb la geometria de la resolució màxima i el tercer pas és acabar d'omplir aquesta taula amb la resta de la geometria gràcies a la relació recursiva.

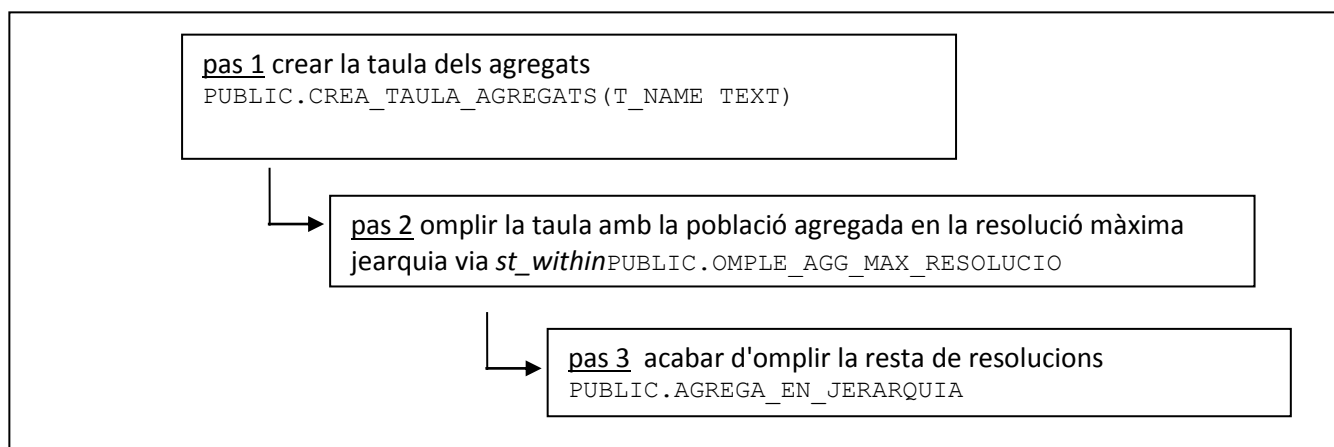


Figura 24 Diagrama del procediment de creació de la informació agregada de la població

7.1.1.1.-Creació de la taula dels agregats

El primer pas és crear la taula dels agregats. Aquesta taula tindrà dues columnes, la columna `ID` que es correspon amb l'id de la geometria de la taula `GRID_ETRS89_LALEA_MULTIRESOL` i la columna `VALOR`, que correspon al valor dels agregats del recompte total de la població.

La taula resultant tindrà el mateix nom que la taula original que conté les dades de tots els elements que cal agrupar però amb l'afegit de "AGG_" al davant.

En el nostre cas, com que la taula que conté tot els elements que cal agrupar és la taula `HABITANTS_EDAT_POSICIO_UTM`, el nom de la taula resultant és `AGG_HABITANTS_EDAT_POSICIO`. El nom de la taula que conté la informació dels agregats és l'únic paràmetre d'entrada (vegeu l'Script de SQL 14)

Aquesta funció també crearà de forma automàtica un índex per a la columna `ID`.

7.1.1.2.-Omplir la taula d'agregats amb la resolució màxima

El segon pas és omplir la taula d'agregats creada en l'apartat anterior amb la resolució màxima.

Aquesta funció, amb el nom `OMPLE_AGG_MAX_RESOLUCIO` necessita tres paràmetres d'entrada. El primer és el nom de la taula on hi ha el valor dels agregats (`T_NAME`), el segon és la resolució màxima (`LEVEL_MAX`) i el tercer és la resolució mínima (`LEVEL_MIN`).

Aquesta funció seleccionarà el valor dels agregats del recompte total de la població que estiguin inclosos en la geometria de màxima resolució. Aquesta operació es fa a partir de la funció de PostGIS `ST_WITHIN`, que opera la posició georeferenciada dels habitants (`POSICIO`) amb la geometria (`GEOM_UTM`) dels quadrats de la resolució màxima (`LEVEL='||LEVEL_MAX||'`). Un cop seleccionats, insereix els valors i l'id en la taula `AGG_HABITANTS_EDAT_POSICIO` (vegeu l'Script de SQL 15).

7.1.1.3.-Omplir la taula d'agregats recursivament

El tercer pas és acabar d'omplir la taula d'agregats amb els valors agregats de població dels nivells superiors del quadtree. Aquest procés es duu a terme gràcies al fet que, tenint el recompte de la població agregada al nivell més baix, podem calcular el recompte de la població agregada als nivells superiors mitjançant la relació de recursivitat dels elements del quadtree.

Aquest pas es fa per mitjà de la funció `AGREGA_EN_JERARQUIA`. Aquesta funció necessita tres paràmetres d'entrada. El primer és el nom de la taula on hi ha el valor dels agregats (`T_NAME`), el segon és la resolució màxima (`LEVEL_MAX`) i el tercer és la resolució mínima (`LEVEL_MIN`).

Aquesta funció fa el recompte (`SUM (VALOR)`) de la població que tenen els quatre quadrats fills de cada quadrat pare, que és el valor de la població del pare. Aquesta funció s'inicia amb un (if) i la condició és que el nivell de resolució sigui igual a la resolució màxima (`CURRENT_LEVEL=LEVEL_MAX`). Un cop calculats, insereix els valors i l'id de la geometria en la taula `AGG_HABITANTS_EDAT_POSICIO`.

Un cop s'acaba de fer la inserció passa a una resolució màxima més alta gràcies al (`CURRENT_LEVEL:=CURRENT_LEVEL+1`). La funció s'acabà quan la resolució màxima és igual a la mínima (`CURRENT_LEVEL = LEVEL_MIN`) (vegeu l'Script de SQL 16).

7.1.1.4.-Encadenament de funcions per a calcular els agregats de la població

Per últim, cal encadenar aquestes funcions en una nova funció per tal de realitzar el conjunt del procés d'obtenció de les dades agregades de tots els nivells del quadtree. Aquesta nova funció necessita els mateixos tres paràmetres que les dues funcions anteriors. El primer és el nom de la taula on hi ha el valor dels agregats (`T_NAME`), el segon és la resolució màxima (`LEVEL_MAX`) i el tercer és la resolució mínima (`LEVEL_MIN`). Aquesta funció conté les altres tres funcions. Executant només aquesta funció s'executen les altres tres.

Amb aquesta funció obtindrem, doncs, els agregats amb el recompte total de la població per els id de totes les geometries de tots els nivells(vegeu l'Script de SQL 17).

7.1.2.-Procediment de creació del quadtree

Un cop tenim els agregats de la població i la geometria, ja es pot crear pròpiament el quadtree.

La creació del quadtree es fa creant primer la taula, després analitzant la geometria partint de la resolució mínima i tot seguit avaluant els seus fills. Si tots els fills estan per sobre del llindar, s'analitza cadascun dels fills d'aquests fills(tornant a iniciar el procés), però si algun dels fills està per sota del llindar, es fa la inserció de la geometria a la taula que hem creat.

Aquest procés s'ha repetit un total de 13 vegades, perquè s'han creat 4 tipus de quadrees segons resolucions màxima i mínima{(62,5_62,5),(62,5_125),(62,5_250),(62,5_500)} amb 4 llindars de població diferents (13,17,19,23). En total, s'haurien de haver creat 16 quadrees, però el quadtree de la resolució de (62,5_62,5) s'ha creat només amb el llindar 17. La justificació recau en el fet que és igual quin llindar s'agafi, perquè a aquesta resolució no hi ha cap agregació de la geometria a nivells superiors, per tant, el llindar no importa a aquesta resolució.

QT	Llindar			
	13	17	19	23
62,5_62,5_RP2014		x		
62,5_125_RP2014	x	x	x	x
62,5_250_RP2014	x	x	x	x
62,5_500_RP2014	x	x	x	x

Figura 25 Resum dels quadrees creats

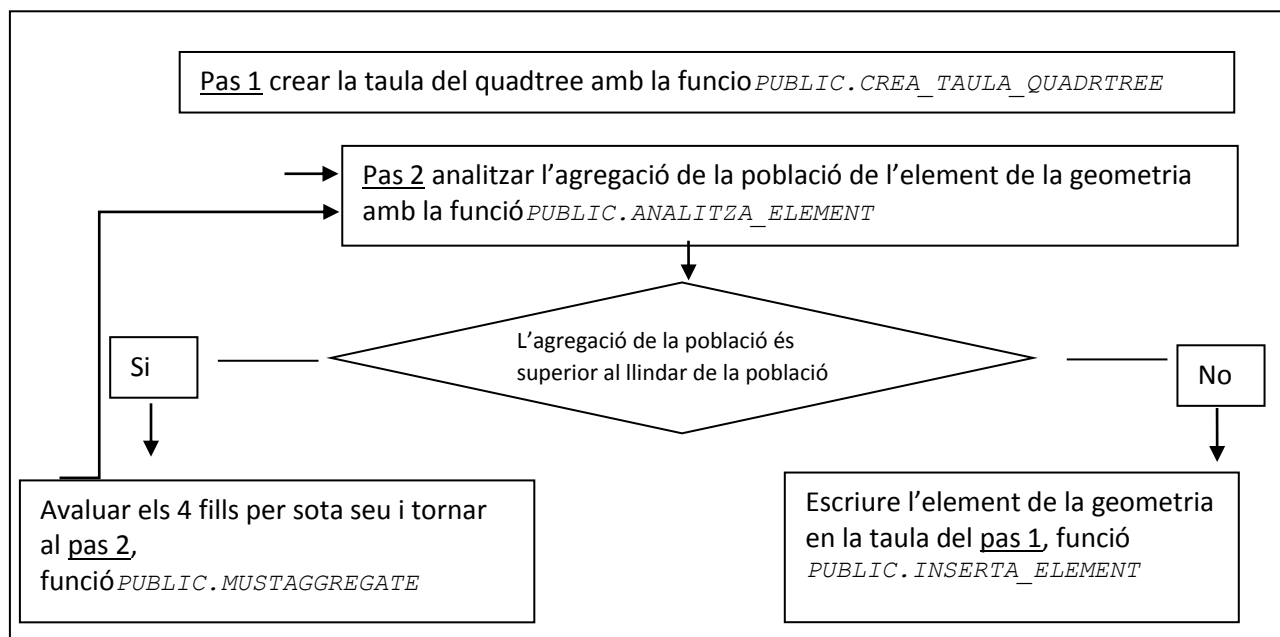


Figura 26 Diagrama del procediment de creació del quadtree

7.1.2.1.-Creació de la taula del quadtree

El primer pas és crear la taula del quadtree. Aquesta taula es crea a partir de la funció CREA_TAULA_QUADRTREE. Aquesta funció necessita quatre paràmetres d'entrada. El primer és la resolució màxima (LEVEL_MAX), el segon és la resolució mínima (LEVEL_MIN), el tercer és el nom de la taula on hi ha el valor de la població, igual que l'apartat anterior (T_NAME) i el quart és el llindar.

El nom de la taula serà la unió de la partícula QT, la resolució màxima, la resolució mínima, el nom de la taula del valor de la població i el llindar (vegeu l'Script de SQL 18).

7.1.2.2.-Anàlisi de cada element geomètric

Un cop hem creat la taula, ja es pot executar la funció que analitza cada element. Aquesta funció és la que fa l'anàlisi sobre cada element del quadtree i, depenent del llindar, escriu la geometria a la taula que hem creat o analitza els seus fills.

La funció es diu ANALITZA_ELEMENT i necessita tres paràmetres d'entrada. El primer és l'ID de la geometria (ID_CERCA), el segon és el llindar (LLINDAR) i el tercer és la taula on es fa la inserció dels resultats (TARGET) .

La funció s'inicia amb la declaració de diferents variables: (RESULT, V, T, CAL_AGREGAR; INDNTIFICADORS) .

El primer que mira la funció és si una altra funció (MUSTAGGREGATE) té un resultat igual a 0. Si el resultat és 0, vol dir que tots els seus fills estan per sobre el llindar i que per tant, cal que torni a inspeccionar els fills dels fills per veure que tots estiguin per sobre del llindar.

Si en algun moment de la iteració, algun dels fills està per sota del llindar, es fa la inserció del'ID del pare a la taula que hem creat a l'apartat anterior (INSERTA_ELEMENT). A continuació es retorna a aquesta mateixa funció per tal que analitzi la resta d'elements.

Aquesta funció per tant, comença sempre per la resolució mínima i va analitzant els fills fins que algun d'aquests està per sota del llindar, llavors escriu el pare (vegeu l'Script de SQL 19).

7.1.2.3.-Anàlisi dels 4 fills de l'element geomètric analitzat

La funció MUSTAGGREGATE està dintre de la funció anterior, i el que fa és mirar tots els fills del mateix pare i comprovar que tots els fills estan per sobre del llindar. Si aquest és el cas, retorna un valor de 0, però si algun dels fills està per sota del llindar, retorna el resultat de -1.

Aquesta funció té dos paràmetres d'entrada. El primer és l'ID de la geometria que analitza (ID_CERCA) i el segon paràmetre és el llindar (LLINDAR).

La funció s'inicia declarant les variables V I T. La variable T és la que posseeix el valor de la població agregada dintre de la geometria (T:=RESULT.VALOR;) i la variable V és la que retorna el valor 0 si tots els fills estan per sobre del llindar o -1 si hi ha algun fill per sota del llindar (vegeu l'Script de SQL 20).

7.1.2.4.-Inserció dels elements geomètrics a la taula

La funció INSERTA_ELEMENT està dintre de la funció ANALITZA_ELEMENT i fa que s'insereixi l'ID a la taula que s'ha creat amb la funció CREA_TAUЛА_QUADTREE.

Aquesta funció té dos paràmetres d'entrada. El primer és l'ID de la geometria que ha d'inserir (ID_INSETAR) i el segon paràmetre és la taula on es fa la inserció (TARGET)(vegeu l'Script de SQL 21).

7.1.2.5.-Encadenament de les funcions per a crear el quadtree

Per últim, com s'ha fet anteriorment, cal concatenar el conjunt de funcions d'aquest procés.

La funció que uneix les funcions anteriors és la funció CREA_QUADTREE i té un total de quatre paràmetres d'entrada, que són els mateixos paràmetres que els del quadtree. El primer és la resolució màxima (LEVEL_MAX) , el segon és la resolució mínima (LEVEL_MIN), el tercer és la taula on hi ha les dades de la població (T_NAME) i el quart és el llindar (LLINDAR).

Aquesta funció comença amb la funció CREA_Taula_QUADTREE i després selecciona un element de la mínima resolució de la taula GRID_ETRS89_LALEA_MULTIRESOL i a continuació comença l'anàlisi dels valors agregats de cada element del quadtree a partir de la funció ANALITZA_ELEMENT (vegeu l'Script de SQL 22).

7.1.3.-Inserció de la informació agregada de la població i de la geometria del quadtree creat

En el pas anterior, s'aconsegueix crear el quadtree, però d'aquest només se sap l'ID de la geometria de GRID_ETRS89_LALEA_MULTIRESOL.

En aquest darrer pas, es reuneix la informació associada a cada element del quadtree (l'id, el nivell, el valor i la geometria).

La informació associada de cada element de la malla dels diferents quadrees s'obté a partir de la unió de tres seleccions. La selecció de la informació de la població agregada per a cada geometria (AGG_HABITANTS_EDAT_POSICIO) amb la selecció de la geometria de malla en els diferents nivells (GRID_ETRS89_LALEA_MULTIRESOL) i amb la selecció de les geometries de malla que formen cada quadtree(QUADTREE_RMAX_RMIN_HABITANTS_EDAT_POSICIO_LLINDAR).

Aquest pas s'ha repetit un total de 13 vegades, una per a cada quadtree creat en l'apartat anterior (vegeu l'Script de SQL 23).

Per últim s'ha creat un índex (b-tree) de la columna del'ID perquè el càlcul dels errors sigui més ràpid.

7.1.4.-Característiques dels quadrees creats

A manera de resum, s'ha calculat el total de quadrats que té cadascun dels quadrees creats. El nombre de quadrats s'ha calculat per a cada resolució.

		Nivell				
		500	250	125	62,5	TOTAL
QT	62,5_62,5_RP2014_17	-	-	-	201589	201589
	62,5_125_RP2014_13	-	-	63333	61831	125164
	62,5_125_RP2014_17	-	-	67416	51772	119188
	62,5_125_RP2014_19	-	-	68928	47886	116814
	62,5_125_RP2014_23	-	-	71283	41716	112999
	62,5_250_RP2014_13	-	24207	17849	54746	96802
	62,5_250_RP2014_17	-	26420	16044	45601	88065
	62,5_250_RP2014_19	-	27253	15292	42106	84651
	62,5_250_RP2014_23	-	28555	14059	36701	79315
	62,5_500_RP2014_13	10798	7075	14395	50007	82275
	62,5_500_RP2014_17	11709	7077	12818	41886	73490
	62,5_500_RP2014_19	12073	7037	12049	38573	69732
	62,5_500_RP2014_23	12692	6823	11012	33583	64110

Figura 27 Nombre de quadrats calculat per a cada resolució

7.2.-Implementació de la pertorbació de coordenades

La implementació de la pertorbació de coordenades s'ha fet en tres passos. El primer pas es la creació dels espais de comparació i de publicació. El segon pas es crear la matriu d'intercanvi on per a cada registre de la població hi hagi els 25 possibles candidats d'intercanvi segons la proximitat. El tercer pas es crear una taula nova (HABITANTS_EDAT_POSICIO_PERTORBAT) on per a cada registre de la població hi hagi el registre amb el qual s'intercanvien els atributs.

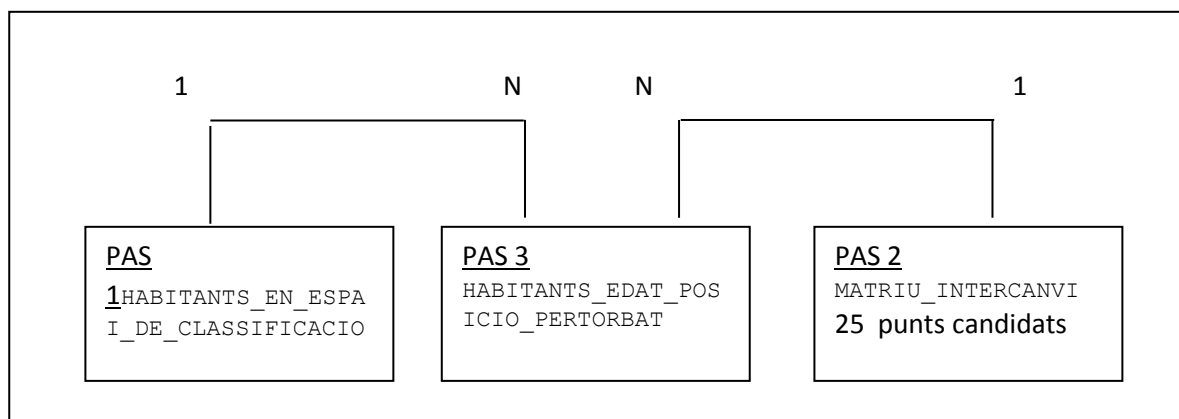


Figura 28 Diagrama d'implementació de la pertorbació de coordenades

7.2.1.-Espai de comparació i espai de publicació

Per a crear l'espai de comparació i l'espai de publicació s'ha creat una taula nova HABITANTS_EN_ESPAI_DE_CLASSIFICACIO. Aquesta nova taula conte tres columnes, l'ID de cada habitant, l'espai de comparació i l'espai de publicació.

Les columnes d'espai de comparació i d'espai de publicació estan omplertes amb un *array*. Aquest *array* es la reclassificació dels atributs de cada habitant amb un codi numèric (sexe, edat, lloc de naixement i codi de província).

En la columna d'espai de comparació, a la primera posició dels *arrays* hi ha l'atribut sexe. El valor d'aquesta primera posició pot ser 1 referent a homes i 2 referent a dones. En la segona posició hi ha l'edat. L'edat en l'espai de comparació està agrupat en grups desenals. Si l'edat està compresa entre 0 i 10 anys (no inclòs), en la segona posició de l'*array* el valor és 1; si l'edat està compresa entre 10 i 20 anys, el valor de la segona posició és 2; etc. En la tercera posició hi ha la nacionalitat, que és 1 si la persona és nascuda a l'estat espanyol i 2 si és nascuda a l'estranger. En la quarta posició hi ha el lloc de naixement, que és 1 per als nascuts a Catalunya, 2 per als nascuts a la resta del estat i 3 per als nascuts a l'estranger (vegeu l'Script de SQL 24).

En la columna d'espai de publicació, la primera, tercera i quarta posició de l'*array* segueix la mateixa estructura que l'anterior. L'únic que canvia és l'edat, ja que en l'espai de publicació l'edat es classifica quinquenal. Per tant, si l'edat de l'habitant està compresa en entre 0 i 15 anys (no inclòs), en la segona posició de l'*array* hi ha 1; si l'edat està compresa entre 15 i 30 anys, en la tercera posició hi ha 2; etc. (vegeu l'Script de SQL 25).

7.2.2.-Implementació de la matriu d'intercanvi

La implementació de la matriu d'intercanvi s'ha fet mitjançant dos passos. El primer pas és aplicar la formula de pertorbació i la segona es construir la matriu d'intercanvi.

La formula de pertorbació és la següent:

$$P(a,b) = \begin{cases} =0 & \text{si } E(a) = E(b) \\ =0 & \text{si } E'(a) <> E'(b) \text{ i } E(a) <> E(b) \\ =f(1/d(a,b)) & \text{si } E'(a)=E'(b) \text{ i } E(a) <> E(b) \text{ i } d(a,b) > 0 \\ =0 & \text{si } E'(a)=E'(b) \text{ i } E(a) <> E(b) \text{ i } d(a,b) = 0 \end{cases}$$

$=0$ si $E(a) = E(b)$ significa que els punts a i b estan dintre del mateix espai de publicació. El fet d'estar dintre del mateix espai de publicació significa que les dues *arrays* són iguals, i que per tant, els valors no es poden intercanviar. L'intercanvi no es pot produir perquè els resultats de l'intercanvi són els mateixos que si no els haguéssim intercanviat.

$=0$ si $E'(a) <> E'(b)$ i $E(a) <> E(b)$ significa que el punt a i el punt b no estan ni en el mateix espai de comparació ni en el mateix espai de publicació. Per tant, els punts no es poden intercanviar perquè tenen atributs massa diferents entre si.

Amb aquesta formula obtindrem valor diferent a 0 quan els atributs dels habitants siguin iguals en l'espai d'intercanvi però diferents en l'espai de publicació sempre que la seva distancia ($d(a,b)$) sigui més gran que zero.

La funció $F(1/d(a,b))$ ens permet escollir aleatòriament un dels possibles candidats fent que sigui més probable l'intercanvi entre punts més propers. Per cada punt s'han cercat els 25 punts més propers que complien les condicions descrites anteriorment (igualtat en l'espai de comparació però no en el de publicació).

Aquesta funció, implica molts càlculs, per això, es va decidir programar aquesta funció amb Java i alhora amb SQL gràcies al *driver* de Java `ORG.POSTGRESQL.DRIVER`. Aquesta funció s'ha fet a partir del MAIN i de les classes `Escriptor_Matriu Java` i `Lector Habitants Java`(vegeu l'Script de Java 1).

El segon pas ha estat escriure la matriu d'intercanvi. La matriu d'intercanvi selecciona de forma aleatòria un dels 25 punts que s'han seleccionat abans. Aquesta selecció aleatòria és la que decideix quin parell de punts s'intercanvien.

Aquesta funció, igual que l'anterior, implica molts càlculs, per això, es va decidir programar aquesta funció amb Java i alhora amb SQL gràcies al *driver* de Java `ORG.POSTGRESQL.DRIVER`. Aquesta funció s'ha fet a partir del MAIN i de les classes `Lector Habitants Java`, `Escriptor Donants` i `Seleccio Donant` (vegeu l'Script de Java2).

7.2.3.-Creació de la taula d'habitants amb la pertorbació de coordenades

Un cop elaborada la matriu d'intercanvi, s'ha efectuat per mitjà d'un altre Script de Java, la transposició de la informació de la matriu anterior a una nova taula amb el canvi de coordenades en les parelles que els correspongui, creant així la pertorbació de coordenades.

Aquesta funció, igual que les dues anteriors, implica molts càlculs, per això, es va decidir programar aquesta funció amb Java i alhora amb SQL gràcies el *driver* de Java `ORG.POSTGRESQL.DRIVER`. Aquesta funció s'ha fet a partir del MAIN i de la classe `Persona` (vegeu l'Script de Java3).

El temps d'execució d'aquest càlcul es molt alt computacionalment i això ha provocat que no haguem tingut temps material per obtenir el resultat del conjunt del total de punts. Tot i així, mostrarem els resultats d'una part que creiem significativa del total: 2.382.186 punts del 7.566.464 punts.

8.-Càlcul dels errors

Per calcular els errors dels quadrees i de la pertorbació de coordenades s'han utilitzat mètodes diferents. En el càlcul d'errors dels quadrees s'ha calculat la distribució dels errors relatius en els experiments, la distribució de les diferències relatives i l'error relatiu segons l'àrea del polígon. L'error de la pertorbació de coordenades s'ha calculat mitjançant la distància de khi quadrat.

8.1.-Polígons de Montecarlo

La taula on hi ha els polígons de Montecarlo es diu MONTECARLO_EXPERIMENTS_POLYGONS i conté tres columnes (el número de l'experiment, l'àrea del polígon i la geometria en sistema de referència ETRS89_UTM_31N).

8.2.-Càlcul dels errors en els quadtree creats

Per a la quantificació de l'error, s'ha començat per fer el càlcul de la població exacta en els polígons generals aleatòriament, després s'ha calculat la població dels polígons en relació als diferents quadrees i per últim s'ha realitzat la comparació de les poblacions anteriors per obtenir els errors relatius.

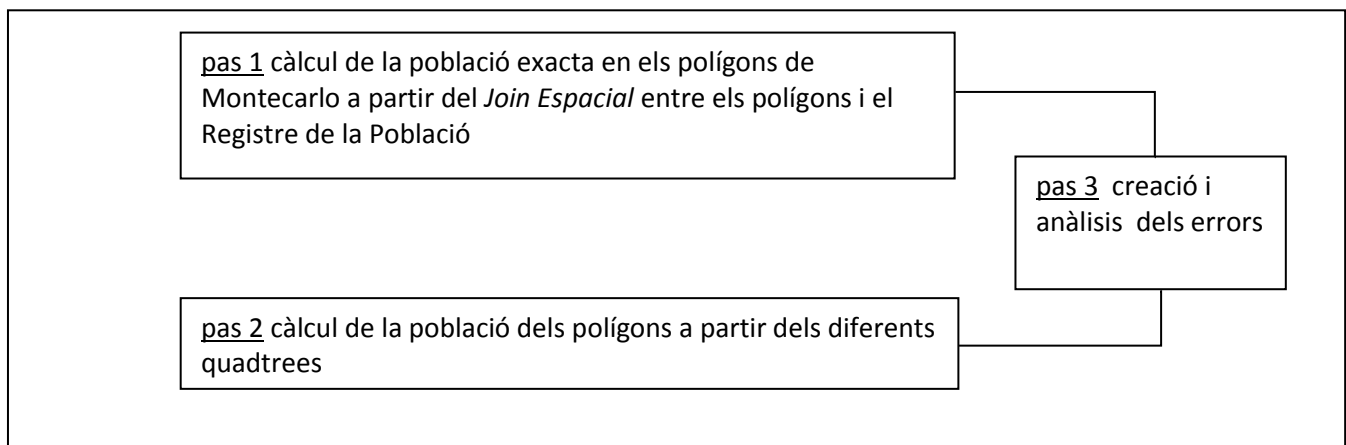


Figura 29 Diagrama del càlcul dels errors en els quadrees creats

8.2.1.-Creació de la taula dels experiments de Montecarlo

Per facilitar l'anàlisi s'ha agrupat la informació en una sola taula. Aquesta taula s'anomena RESULTAT_EXPERIMENT_MONTECARLO i conté un total de 15 columnes. La primera columna és el número de l'experiment, la segona és la població exacta que inclou el polígon i les tretze columnes posteriors és la població segons cada un dels quadrees creats.

Els quadrees amb una resolució màxima i mínima de nivell 4 (62.5m) { Qt(4,4,RP14,13),Qt (4,4,RP14,17),Qt (4,4,RP14,19) i Qt (4,4,RP14,23) } tindran el mateix error, perquè en aquests no hi ha cap agregació. Així, per evitar redundància, s'eliminen les columnes amb els llistats 13,19 i 23 d'aquest quadree ja que amb la columna POB_4_4_17 es resumeix tota la informació(vegeu l'Script de SQL 26)

8.2.2.-Càlcul de la població exacta dels polígons

En el càlcul de la població exacta que hi ha dintre dels polígons creats, s'ha realitzat una intersecció espacial entre la taula MONTECARLO_EXPERIMENTS_POLYGONS i la taula HABITANTS_EDAT_POSICIO(vegeu l'Script de SQL 27).

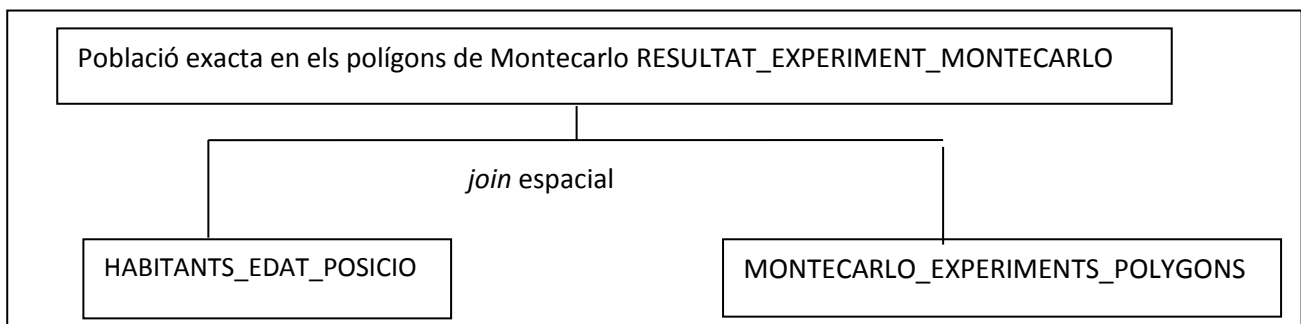


Figura 30 Diagrama del càlcul de població exacta ens els polígons de Montecarlo

El resultat d'aquest *join espacial* s'emmagatzema dintre de la segona columna de la taula RESULTAT_EXPERIMENT_MONTECARLO.

8.2.3.-Càlcul de la població segons cada quadree

Per a calcular la població de cada polígon segons cada quadree s'ha efectuat una intersecció espacial entre la taula MONTECARLO_EXPERIMENTS_POLYGONS i cada una de les 13 taules QT...creades en l'apartat 8.1.2.

El resultat obtingut en cadascuna de les 13 interseccions s'ha guardat a la columna corresponent de la taula RESULTAT_EXPERIMENT_MONTECARLO segons el quadree que s'ha utilitzat. Aquest procés el fem a partir d'un *UPDATE*. En l'script de SQL 28 es mostra l'exemple del codi de SQL per a obtenir la població del Qt(4,3,RP2014,13), que es guarda dintre de la columna POB_4_3_13.

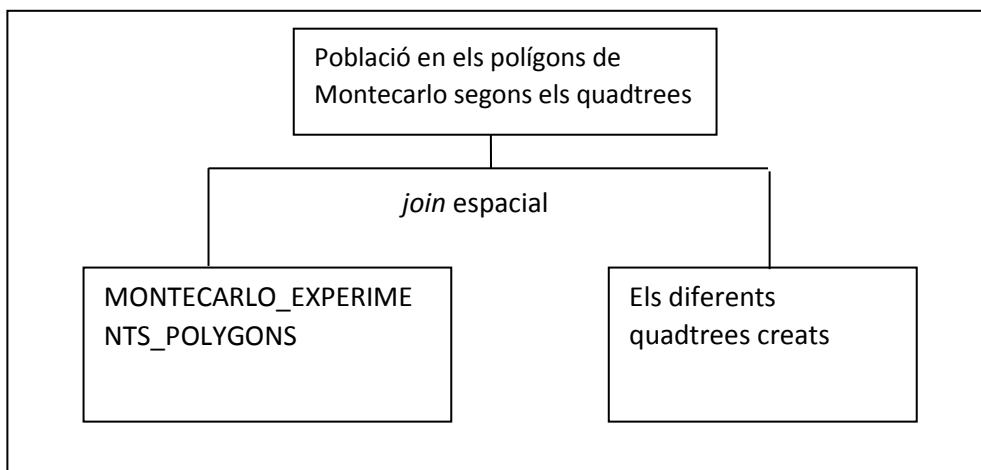


Figura 31 Diagrama del càlcul de població segons cada quadtree

8.2.4.-Càlcul dels errors

Els càlculs dels errors s'han realitzat a partir de tres estudis estadístics. El primer és la distribució de l'error relatiu dels experiments. El segon és la distribució de les diferències relatives. El tercer és la distribució de l'error relatiu segons l'àrea dels polígons dels experiments.

Cada estudi estadístic s'ha encarat amb dos enfocaments. El primer enfocament és analitzar els errors en cadascun dels quadrees creats per saber com es comporta l'error segons els paràmetres del quadtree. L'altre enfocament és veure com es comporten aquests errors en els diferents quadrees d'un mateix llinar i així esbrinar quins són els paràmetres més indicats per a la construcció del quadtree.

8.2.5.-Distribució de l'error relatiu en els experiments

Aquest error es calcula a partir de l'obtenció del valor absolut de la diferència entre la població exacta en els polígons i la població segons els quadrees dividida per la població exacta.

Aquest càlcul s'ha repetit 13 vegades (una per a cada columna de la taula RESULTAT_EXPERIMENT_MONTECARLO) i els resultats s'han agrupat segon els llinars dels quadrees creats. Aquests resultats s'han exportat a .CSV per una major facilitat en el tractament estadístic posterior(vegeu l'Script de SQL 29).

Per a analitzar la distribució de l'error relatiu, s'han calculat els paràmetres de distribució central d'aquest, en concret la mitjana i la mediana. Aquesta distribució central queda resumida en un diagrama de caixa. Els diagrames de caixa es mostren a la figura 32.

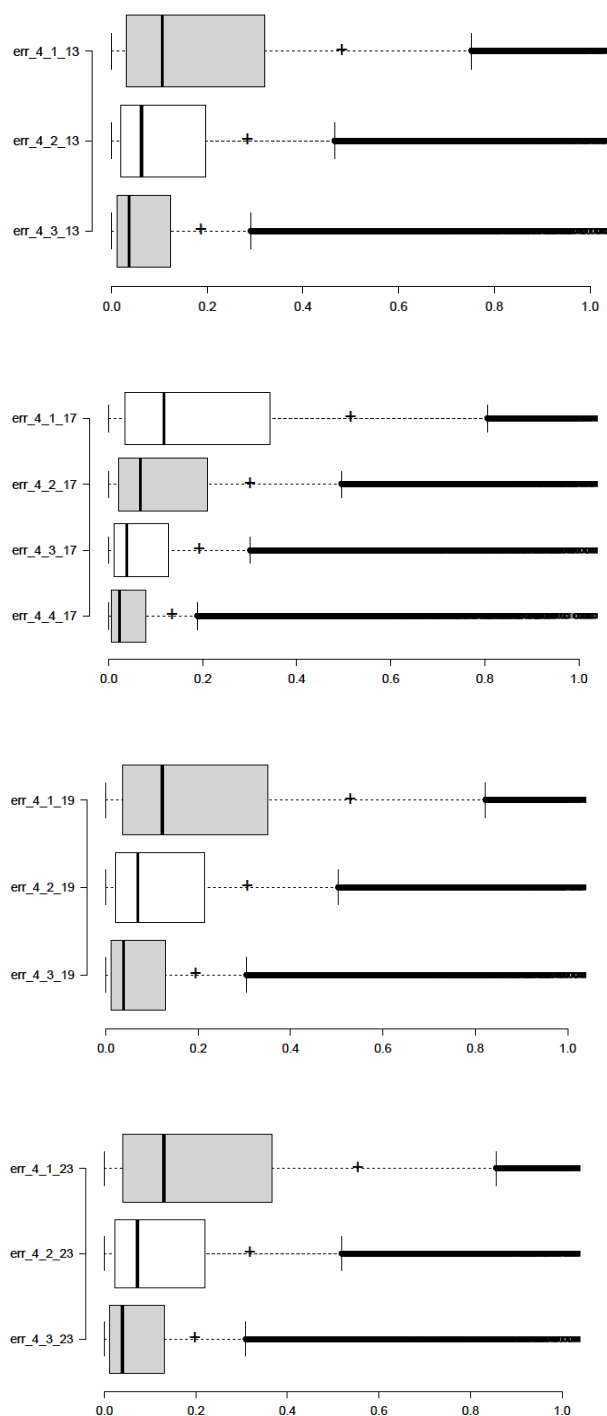


Figura 32 Diagrames de caixa dels diferents quadrees

En els diagrames anteriors es pot observar que la mitjana és molt superior a la mediana i que en tots els diagrames hi ha un gran nombre de casos atípics (*outliers*), motiu que fa que la mitjana sigui molt superior a la mediana. També es pot observar que a mesura que augmenta la resolució mínima augmenten els valors de la mitjana i la mediana. Per tant, com menor és la resolució mínima menors són els errors.

Aquesta relació entre la resolució mínima i els valors de la mitjana i la mediana es veu clarament si representem els valors de la mediana (figura 33) i la mitjana (figura 34) dels errors en funció de la resolució.

La mediana de la distribució de l'error relatiu augmenta de forma lineal. Quan augmenta la resolució mínima, augmenta l'error. Però els quadrees amb un llindar 13 tenen un error força més baix comparat amb els quadrees amb un llindar 23. Això mostra que el llindar influeix en l'error del quadree, però menys que la resolució mínima.

La mitjana de la distribució de l'error relatiu també augmenta de forma lineal, i segueix el mateix patró que en el cas de la mediana i també es pot observar com quan més baix és el llindar, l'error de la mitjana també és més baix.

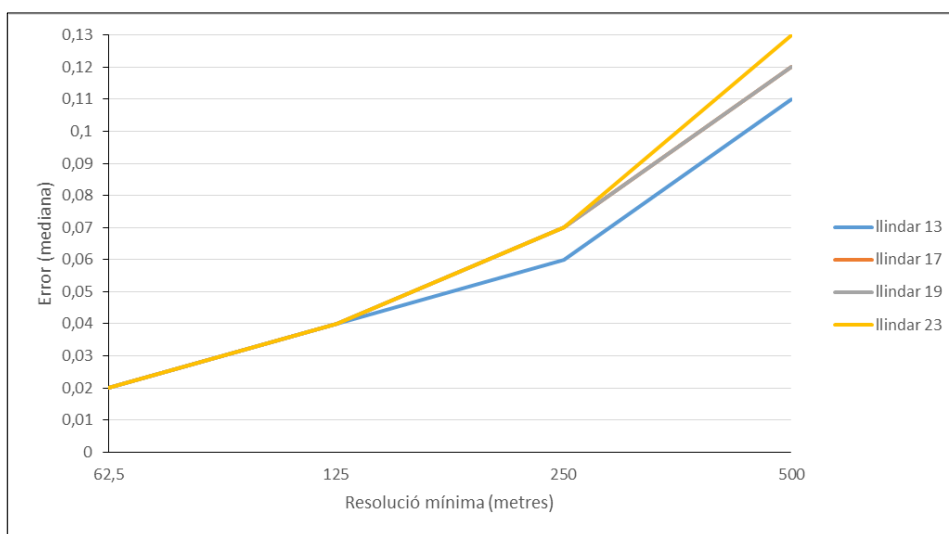


Figura 33 Distribució de l'error (mediana) respecte del llindar amb la resolució mínima

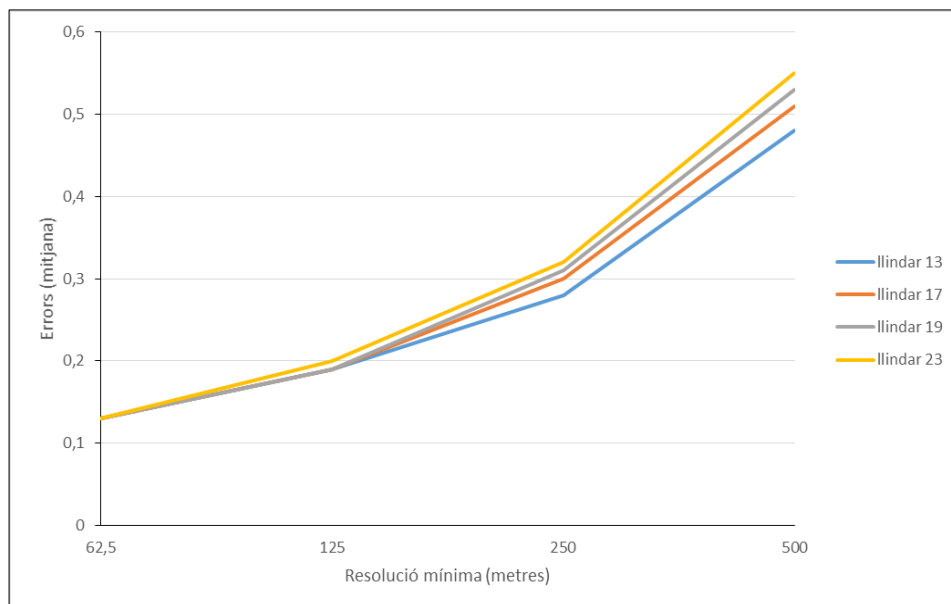
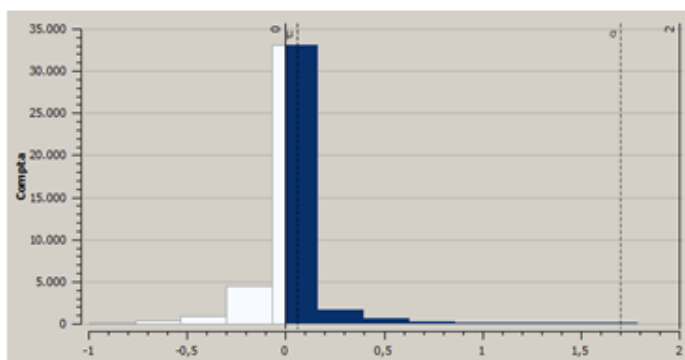


Figura 34 Distribució de l'error (mitjana) respecte del llindar amb la resolució mínima

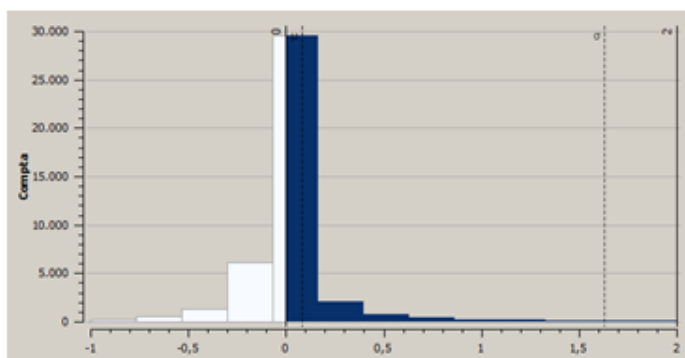
8.2.6.-Distribució de les diferències relatives

La distribució de les diferències relatives ens mostra si els quadrees creats estan sobreestimant la població o al contrari. Aquesta distribució s'ha realitzat a partir de la resta de la població segons el quadtree amb la població exacta dividida per la població exacta (vegeu l'Script de SQL 29).

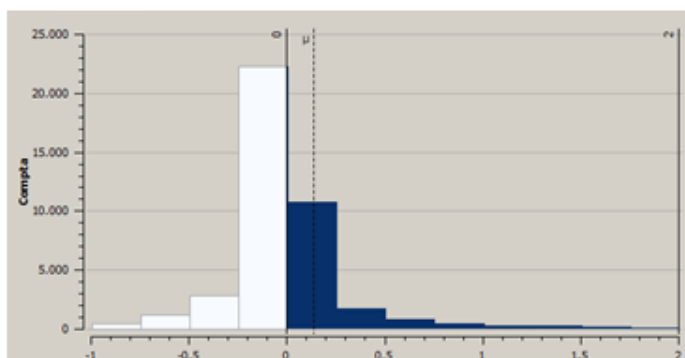
S'ha elaborat un histograma amb les dades anteriors per veure la distribució de les diferències relatives segons el quadtree utilitzat. En color blau es representen els valors que estan per sobre de 0 i que per tant estan sobreestimant la població i en color blanc els que són inferiors a 0 i que per tant subestimen la població.



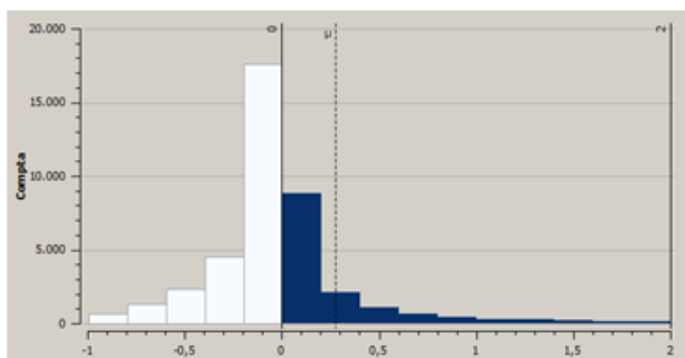
QT{62.5m,62.5m,RP2014,17}



QT{62.5m,125m,RP2014,17}



QT{62.5m,250m,RP2014,17}



QT{62.5m,500m,RP2014,17}

Figura 35 Histograma de la distribució de les diferències relatives

Com es pot observar la mitjana de les diferències està per sobre de 0, la qual cosa significa que hi ha una tendència a sobreestimar la població de les àrees d'interès quan es calcula a partir dels valors de població dels quadrees. En els últims histogrames hi ha un gran nombre de casos negatius (part esquerra, color blanc), però la mitjana continua sent positiva, per tant hi ha una tendència a sobreestimar la població.

Si comparem els diferents histogrames, es pot observar com la línia de la mitjana es va desplaçant cap a la dreta com més gran és la resolució mínima. Això significa que l'error és més gran quan l'agregació és més gran (resolució mínima més gran implica quadrats més grans). En els QT{62.5m,62.5m,RP2014,17} i QT{62.5m,125m,RP2014,17}, la mitjana està en uns valors força petits però en el QT{62.5m,250m,RP2014,17} i sobretot en el QT{62.5m,500m,RP2014,17} la mitjana de l'error és molt gran. Per tant, s'arriba a la conclusió que com menor sigui la resolució mínima, menys possibilitats d'error hi ha en el càlcul de la població d'àrees d'interès fetes per l'usuari a partir dels valors de població facilitats per mitjà dels quadrees.

8.2.7.-L'error relatiu segons l'àrea del polígon

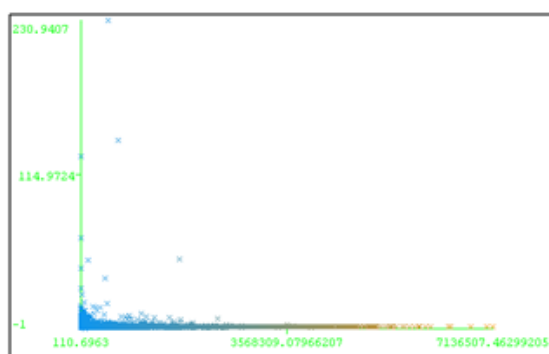
En aquesta anàlisi s'ha avaluat l'error relatiu en relació a l'àrea del polígon. Per a representar aquesta possible relació s'ha creat un diagrama de dispersió de punts. La figura 36 mostra aquests diagrames per als quadrees de l'indar 17.

En el diagrama de dispersió, l'eix de la x representa l'àrea en metres dels polígons de Montecarlo i l'eix de la y l'error relatiu.

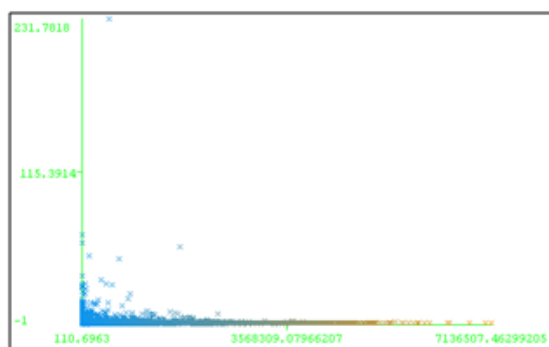
En aquesta figura es pot observar que en tots els diagrames, com més petites són les àrees dels polígons, més alts són els nivells d'error. Aquest fet s'explica pel fet que, com que partim del supòsit que dins del quadtree la població es distribueix uniformement per tota la geometria de cada quadrat (cosa que en la majoria de casos és errònia), i els polígons a causa de la seva forma i posició tenen un gran nombre d'interseccions amb el quadtree, una superfície petita del polígon provoca que l'error augmenti.

Lligada a la idea anterior, s'observa que els errors també són més alts com més gran és la resolució mínima. Aquest fet es demostra comparant el QT{62.5m,500m,RP2014,17} i el QT{62.5m,62.5m,RP2014,17}. El quadtree amb una resolució mínima de 500m té una dispersió d'error molt més alta que el quadtree amb una resolució mínima de 62.5m.

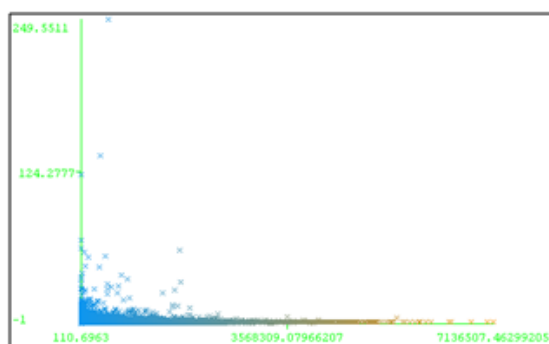
També es pot observar com hi ha un gran nombre de casos atípics, i que aquests són deguts als polígons amb unes dimensions més petites, perquè aquests tenen més interseccions amb la geometria del quadtree.



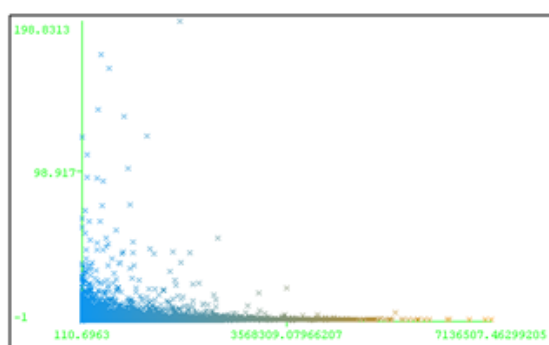
QT{62.5m, 62.5m, RP2014, 17}



QT{62.5m, 125m, RP2014, 17}



QT{62.5m, 250m, RP2014, 17}



QT{62.5m, 500m, RP2014, 17}

Figura 36 Diagrames de dispersió de l'error relatiu en relació a l'àrea del polígon amb llindar 17

Si analitzem el cas més atípic de tots (encerclat en color vermell a la figura 37) veiem com aquest polígon té l'error tan elevat a causa de la intersecció amb el quadtree i el nombre d'elements del quadtree que inclou parcialment.

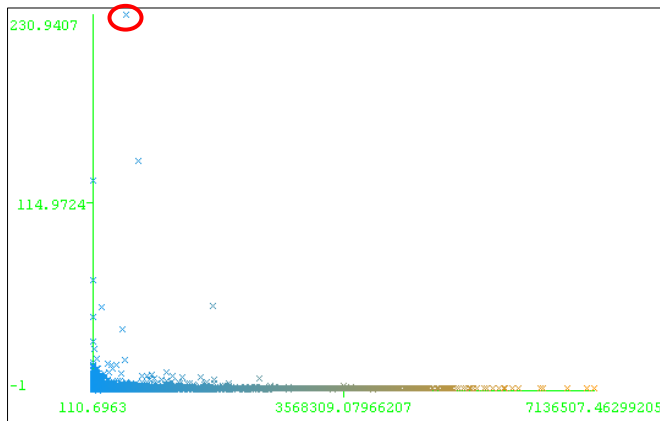


Figura 37 Error relatiu d'un cas atípic segons el quadtree {62.5m,62.5m,RP14,17}

Aquest cas té l'error tan gran perquè intersecciona només amb un sol element de la malla del quadtree. A la figura 38 es pot veure el polígon de Montecarlo i la seva intersecció amb el quadtree QT{62.5m,62.5m m,RP2014,17}. En aquest cas, l'error relatiu és $\epsilon=230.94$

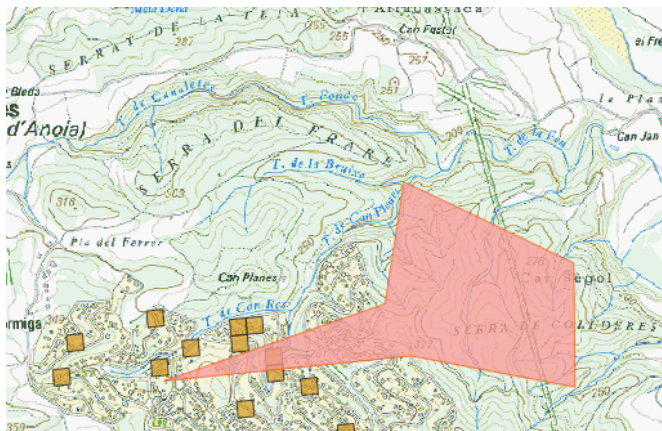


Figura 38 Error relatiu del cas atípic anterior segons el quadtree {62.5m,62.5m,RP14,17}

Derivat de la idea anterior, veiem que en la figura posterior (Figura 39) s'observa com la intersecció dels polígons amb el quadtree augmenta l'error. En el primer cas, el polígon té una intersecció completa, per tant, l'error és de 0. En canvi, en el segon cas, tot i ser el mateix quadtree, l'error augmenta a causa de la petita intersecció donada per la forma del polígon. En aquest cas, l'error és de 0.0006.

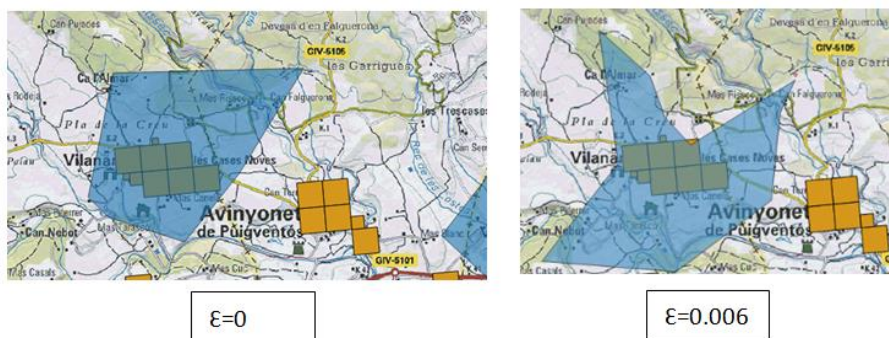


Figura 39 Errors relatius extrems inferiors per al quadtree{62.5m,250m,RP2014,17}

8.3.-Errors de la pertorbació de coordenades

Per mesurar els efectes de la pertorbació, hem utilitzat el conjunt de 50.000 polígons aleatoris utilitzats en els experiments de Montecarlo anteriors. Sobre cada un d'aquests polígons hem calculat la distribució de les edats en l'espai de publicació, obtenint un vector de dimensió 19 tant per la capa original com per a la pertorbada(vegeu l'Script de SQL 31).

8.3.1.- Anàlisi de khi quadrat

Per a mesurar la semblança entre els dos vectors hem utilitzat la distància de khi quadrat sobre els vectors normalitzats.

La distribució de les distàncies, en els casos en què hi ha població dins els polígons (40.171) és:

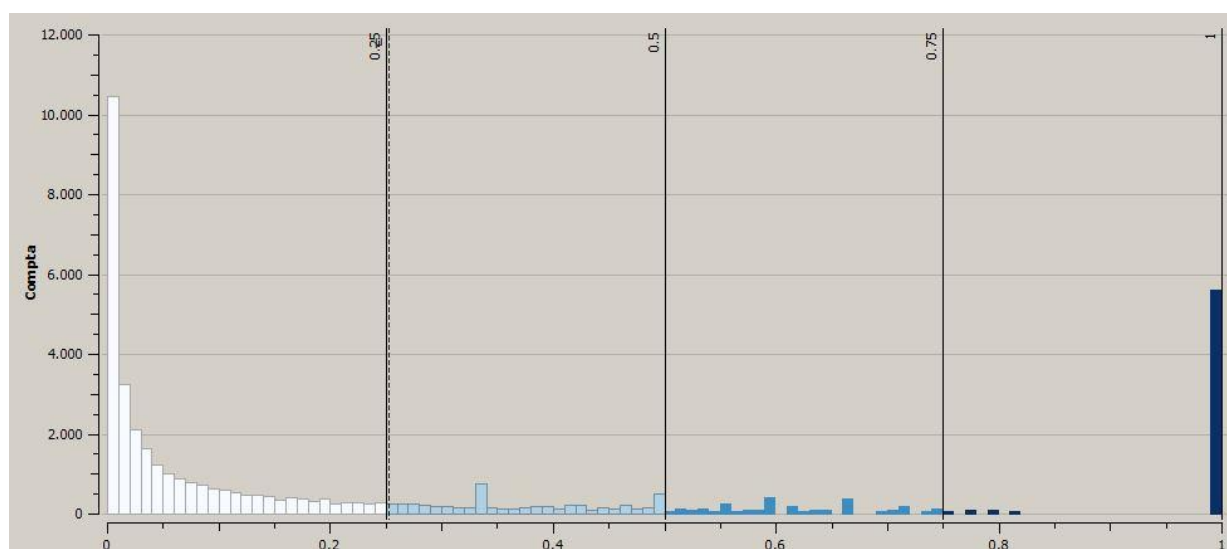


Figura 40 Distància de khi quadrat sobre els vectors normalitzats

Com es pot observar la gran majoria de casos presenten una distancia petita, tot i que apareixen *outliers* i hi ha un pic notable en el valor 1. Els casos que tenen distancia 1 són totalment diferents. Es corresponen a casos en què el total de població és extremadament petit (tretze persones o menys) i la seva freqüència és de 4481 (aproximadament un 10% del total de polígons).

El diagrama de dispersió entre la distancia de khi quadrat i la població dins els polígons es mostra a la figura 41.

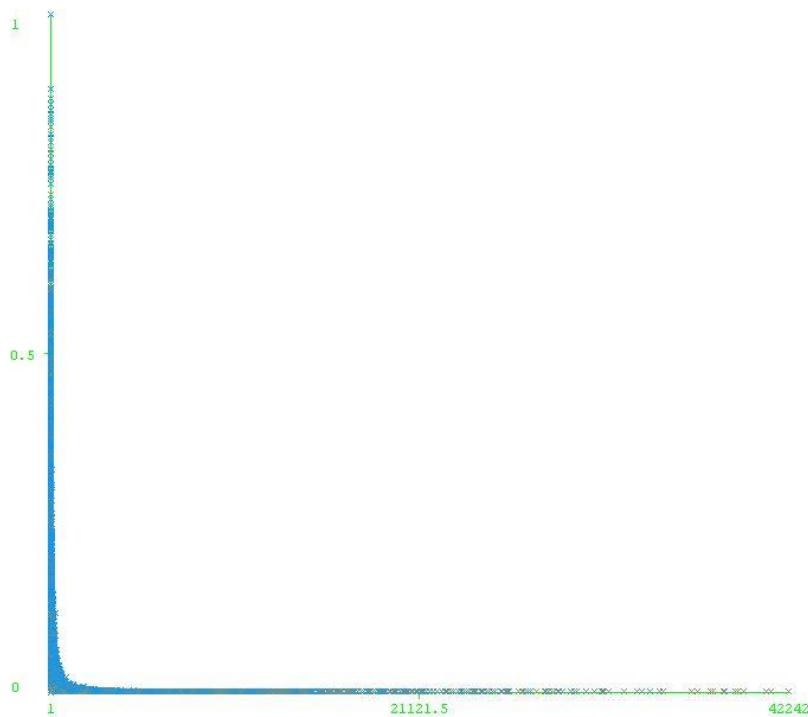


Figura 41 Diagrama de dispersió entre la distancia de khi quadrat i la població

Tot i que el resultat és parcial veiem que per als polígons on hi ha prou població la distancia tendeix ràpidament a zero i que sols és gran en el cas en què la població dins els polígons és baixa. Per últim cal fer constar que per a qualsevol càlcul que no inclogui la variable edat (que controla la pertorbació) l'error es zero necessàriament. Un cop Idescat tingui els càlculs acabats per al total d'habitants completarà l'anàlisi d'errors i avaluarà els riscos de vulneració del secret estadístic en la revelació del resultat pertorbat.

9.-Conclusions

Les conclusions s'han organitzat en quatre punts. Les conclusions sobre els objectius, que resumeix la valoració extreta de l'anàlisi d'errors dels quadrees i de la pertorbació de coordenades. Les conclusions metodològiques, on s'expliquen possibles tècniques metodològiques per a estudis pròxims relacionats amb els quadrees i amb la pertorbació de coordenades. Les conclusions sobre aplicacions futures de les dades, on s'expliquen quins són els possibles estudis o anàlisis que es poden fer amb les dades elaborades al llarg d'aquest treball.

9.1.-Conclusions sobre els objectius

Les conclusions sobre els objectius s'han estructurat en dos punts. El primer punt són les conclusions sobre els quadrees on s'expliquen els paràmetres més òptims per a la construcció del quadree i el quadree més òptim construït. El segon punt són les conclusions sobre la metodologia de pertorbació de coordenades.

9.1.1.-Paràmetres del quadree

Un cop acabada l'anàlisi dels errors dels quadrees, s'han extret les següents conclusions sobre quins són els paràmetres més idonis i per tant, quin és el quadreemillor construït.

Així doncs, les conclusions en aquest aspecte tracten sobre la resolució màxima, la resolució mínima i el llinar més òptims.

9.1.1.1.-Resolució màxima

Com s'ha vist en l'apartat 6.2.5, els quadrats del quadree de 62,5 metres, tenen una població agregada per sobre del llinar en les principals zones urbanes de Catalunya (Barcelona, Sabadell, Terrassa, Lleida, Girona, entre d'altres...). El fet que en aquestes ciutats més poblades i amb més habitants de Catalunya, la geometria de 62,5 metres tingui una agregació de la població per sobre del llinar (17), demostra que la resolució de 62,5 metres és més òptima que la geometria de 125 metres.

És més òptima perquè té un major detall que la geometria de 125 metres, per tant, es pot concloure que la resolució màxima ha de ser de 62,5 metres.

9.1.1.2.-Resolució mínima

La resolució mínima pot ser de 62,5 metres , de 125 metres , de 250 metres i de 500 metres.

En la taula següent (Figura 42), veiem com el % de població nul·la és molt elevat (9,8%) en el quadtree de resolució mínima de 62,5 metres. Per tant, es pot descartar el valor de 62,5 metres com a resolució mínima.

En la mateixa taula, s'observa que el quadtree amb una resolució mínima de 125 metres té una població nul·la del 3,23%, mentre que en el de 250 metres és d'1,16 %. Per tant es pot concloure que la resolució mínima de 250 metres és més òptima que la de 125 metres.

La resolució de 500 metres presenta un % de la població molt alt, però no podem seleccionar aquesta resolució ja que en la Figura 33 i 34 es veu com la resolució de 500 presenta un alt error relatiu. Aquest fet demostra que la resolució de 250 metres suposa un equilibri més òptim pel que fa al % de la població i l'error relatiu, mentre que en la resolució de 500 no hi ha aquest equilibri.

62,5m *_RP2014_17			
nivell	total de la població	% Població nul·la	% Població
62,5m	741264	9,80	90,20
125m	249028	3,29	96,71
250m	87678	1,16	98,84
500m	41917	0,55	99,45
Població total	7566464		

Figura 42 Taula de comparació de la població nul·la en els diferents quadrees amb llinar 17

En resum, la resolució màxima ha de ser de 250 metres.

9.1.1.3.-Llinar

El llinar és el paràmetre que més costa d'analitzar per a trobar el valor més òptim, ja que com s'ha vist en l'apartat 3, no hi ha una metodologia clara de com arribar a trobar el valor més òptim.

Per arribar a trobar el llinar més òptim, s'ha fet una taula resum on es compara la població nul·la en els quadrees amb una resolució màxima de 62,5 metres i una resolució mínima de 250 metres.

Resum llinar		
QUADTREE	Població nul·la	% Població nul·la
62,5_250_RP2014_13	64831	0,86
62,5_250_RP2014_17	87678	1,16
62,5_250_RP2014_19	99683	1,32
62,5_250_RP2014_23	123929	1,64

Figura 43 Taula resum del llinar en els quadrees 62,5_250_RP2014

En aquesta taula es pot veure que el % de la població nul·la augmenta quan augmenta el llindar. L'ideal seria agafar el quadtree amb un % de població nul·la més baix, però si agafem un llindar de 13, posaríem en risc la confidencialitat estadística de les dades.

Davant d'aquesta problemàtica, s'ha decidit seleccionar el llindar corresponent al següent % de població nul·la més baix(1,16%), és a dir un llindar de 17 habitants. Amb aquest llindar, s'assegura que no hi ha risc de vulneració del secret estadístic i dona una població nul·la força més baixa en comparació amb el llindar 23 i amb el llindar 19.

Per concloure, el llindar més òptim és el llindar 17.

9.1.2.-El quadtree idoni

A tall de síntesi, s'ha volgut crear una taula resum on apareguin les principals característiques del quadtree idoni.

Aquest quadtree és el que té una resolució màxima del 62,5 metres, una resolució mínima de 250 metres i un llindar de 17.

62,5_250_RP2014_17			
nivell	Recomptequadrats	Quadratsnuls	% Quadratsnuls
62,5m	45601	-	
125m	16044	-	
250m	26420	15072	57,05
TOTAL	88065	15072	17,11

Figura 44 Taula resum del recompte de quadrats i de quadrats nuls

Com es pot veure en aquesta taula, només queden nuls un 17% del total de quadrats del quadtree. Aquests quadrats nuls, com és lògic, són de la resolució mínima (250 metres) i representen el 57% de quadrats d'aquesta resolució.

Aquests quadrats nuls, representen doncs el 1,16% de la població total de Catalunya (Registre de Població de l'any 2014).

62,5_250_RP2014_17			
nivell	total de la població	Població nul·la	% Població nul·la
62,5m	4886367		
125m	1471819		
250m	1208257	87678	1,16
TOTAL	7566443		

Figura 45 Taula de la població nul·la en el quadtree 62,5_250_RP2014_17

9.1.2.- Pertorbació de coordenades

Un cop acabada l'anàlisi de la pertorbació de coordenades, veiem que els resultats són semblants al resultat que obtenim amb els quadrees. Veiem que els polígons a on hi ha força població la distància tendeix a zero. Això significa que la pertorbació de coordenades és una metodologia molt òptima per a l'equilibri entre el secret estadístic i l'alta resolució espacial.

També es pot veure que els polígons on hi ha poca població mostren una distància d'error molt alta. Aquest fet també és força semblant a l'error dels quadrees.

Però aquest error només afecta l'edat dels habitants, ja que per a la resta de variables (sexe, nacionalitat i lloc de naixement) l'error és zero. Aquest fet sumat a que la resolució espacial de la pertorbació de coordenades és molt més alta que la dels quadrees, ja que en la pertorbació de coordenades continuem tenint una capa de punts(habitants individuals), es pot concloure que la pertorbació de coordenades és el millor sistema de representació espacial de les dades de la població sense que afecti el secret estadístic.

Cal recordar, però, que aquestes conclusions així com l'anàlisi no s'ha fet amb el total de la població sinó que s'ha fet amb una part representativa d'aquesta. Aquest fet pot afectar en l'error de la pertorbació ja que potser amb la totalitat de punts, els errors poden baixar subtilment.

Però també cal remarcar que aquesta metodologia ha estat un prova o una aproximació a una metodologia real, ja que Idescat encara ha de revisar els resultats finals (amb tota la població) per indicar si realment es preserva el secret estadístic o no.

9.2.-Conclusions sobre la metodologia

Les conclusions sobre la metodologia es centren en la determinació del llindar en el cas de l'elaboració dels quadrees i en la determinació de pertorbar les coordenades segons l'edat i no per altres atributs com el sexe i la nacionalitat.

Tal i com s'ha comentat (6.1), no hi ha una metodologia per la cerca del llindar més òptim. Això fa que s'hagin de seleccionar diferents llindars i analitzar-los per descobrir quin és el llindar més òptim. Aquesta anàlisi empírica i poc precisa pot provocar que el llindar seleccionat per a crear els diferents quadrees sigui poc encertat. Per tant, caldria establir una metodologia clara i eficient per a trobar aquests llindars.

La decisió de pertorbar sols per l'edat és per que qualsevol càlcul amb altres variables és exacte i al ser la edat una variable continua la pertorbació es pot ajustar, però com s'ha demostrat en la implementació, la pertorbació de coordenades per l'edat és un procés força senzill des del punt de vista resolutiu i a més presenta uns errors baixos en els polígons on hi ha un cert nombre de població.

9.3.-Conclusions sobre futures aplicacions de les dades

Les conclusions sobre futures aplicacions se centren en els futurs estudis que es poden fer gràcies al conjunt de dades creades en aquest treball.

La creació de funcions per a la implementació dels quadrees provoca que si hi ha algun altre tècnic d'Idescat que vulgui crear un quadree a partir d'altres dades puntuals com per exemple establiments... pugui fer-ho sense tornar a iniciar tot el procés de metodologia, ja que només hauria de canviar els paràmetres d'entrada de les funcions creades. Així mateix, si en un futur Idescat decideix canviar els valors dels paràmetres i tornar a crear nous quadrees, només ha de canviar els valors dels paràmetres a la funcions creades.

Ja sigui amb la representació a partir de quadrees o amb la de pertorbació de coordenades, ambdues metodologies permeten obtenir dades adequades per a fer estudis amb molt detall d'àrees o de zones d'interès que abans eren impossibles de fer. Per tant, les dades d'aquest treball serveixen de base per a diferents projectes acadèmics o de consulta més simple dels usuaris i tècnics.

Aquest treball serveix per obrir una mica el ventall de la representació espacial mitjançant sistemes d'informació geografia de les dades georeferenciades espacialment. Les dues metodologies poden servir de base per a la creació de metodologies més complexes en un futur.

10.-Referències

Aluru, S. (2005). Quadrees and Octrees. In S. S. In D. Metha, *Handbook of Data Structures and Applications* (pp. 280-392). Chapman & Hall/CRC.

Consideracions sobre la utilització de quadrees en la difusió de dats geocodificats y la preservació del secret estadístic. 2016, *X Jornadas de Estadística Pública*

David Eppstein et al. (2005). The Skip Quadtree: A Simple Dynamic Data Structure for Multidimensional Data. *Cornell University Library*.

Lagonigro Raymond et al. (2016) A quadtree approach based on European geographic grids: reconciling data privacy and accuracy. *Data Analysis and Modeling Research Group*

Mark de Berg et al. (2008). *Computational Geometry*. Springer.

Martin Behnisch et al. (2013) Using quadtree representation in building stock visualization and analysis. *Erdkunde Vol. 67 · No. 2* 151–166

R. A. Finkel et al. (1974). Quad trees a data structure for retrieval on composite keys. *Springer-Verlag*.

Suñé, E. (2015). Hacia un registro estadístico del territorio. *Servei de sistemes d'informació geogràfica i teledetecció*.

11.-Annexos

Script de SQL 1 Creació de la taula HABITANTS_EDAT_POSICIO

```
CREATE TABLE HABITANTS_EDAT_POSICIO_UTM
```

```
AS SELECT
```

```
GID ,
```

```
PRO ,
```

```
MUN ,
```

```
DIS,
```

```
SEC ,
```

```
CPRON,
```

```
CMUNN,
```

```
EDAT,
```

```
SEXO,
```

```
NACI,
```

```
ID_FONT ,
```

```
D_QUALITAT ,
```

```
POSICIO ,
```

```
FK_GRID250
```

```
FROM HABITANTS_EDAT_POSICIO
```

Script de SQL 2 Intersecció de les dades de població amb les geometries de tots els nivells del quadtree

```
CREATE TABLE POBLACIO_DINS_CUADRATS AS
SELECT ID, COUNT(*) AS POBLACIO
FROM GRID_ETRS89_LALEA_MULTIRESOL,HABITANTS_EDAT_POSICIO_UTM
WHERE ST_INTERSECTS(POSICIO,GEOM_UTM)
GROUP BY ID
```

Script de SQL 3 Càlcul de les freqüències de la població en la resolució espacial de 1000m

COPY

```
(SELECT POBLACIO,COUNT(*) AS FREQ,COUNT(*)*POBLACIO AS ACUM
FROM GRID_ETRS89_LALEA_MULTIRESOL B,POBLACIO_DINS_CUADRATS A
WHERE A.ID=B.ID AND POBLACIO > 0 AND POBLACIO < 40 AND LEVEL=0
GROUP BY POBLACIO ORDER BY POBLACIO ASC)
TO 'D:/ANALISI/LLINDAR_LEVEL0.CSV' CSV HEADER DELIMITER ','
```

Script de SQL 4 Funció DIVIDE

```
CREATE OR REPLACE FUNCTION PUBLIC.DIVIDE (
INTEGER,
    CHARACTER VARYING,
    CHARACTER VARYING,
    CHARACTER VARYING,
    DOUBLE PRECISION,
    DOUBLE PRECISION,
    DOUBLE PRECISION,
    DOUBLE PRECISION,
    INTEGER)
RETURNS INTEGER AS
$BODY$DECLARE
    GID ALIAS FOR $1;
    GRD_FIXID ALIAS FOR $2;
    GRD_FLOAID ALIAS FOR $3;
    GRD_NEWID ALIAS FOR $4;
    X_ORIGEN ALIAS FOR $5;
    Y_ORIGEN ALIAS FOR $6;
    MAX_RES ALIAS FOR $7;
    AMPLADA_INICIAL ALIAS FOR $8;
    LEVEL ALIAS FOR $9;

    XMIN DOUBLE PRECISION;
    XMAX DOUBLE PRECISION;
    YMIN DOUBLE PRECISION;
    YMAX DOUBLE PRECISION;
    AMPLADA DOUBLE PRECISION;

    CONT INTEGER;
    CURRX DOUBLE PRECISION;
    CURRY DOUBLE PRECISION;
    CONTOT INTEGER;
    GEOMETRIA GEOMETRY;

BEGIN
    CONTOT:=0;
    XMIN:=X_ORIGEN;
    XMAX:=X_ORIGEN+AMPLADA_INICIAL;

    YMIN:=Y_ORIGEN;
    YMAX:=Y_ORIGEN+AMPLADA_INICIAL;

    AMPLADA:=AMPLADA_INICIAL/2.;

    CURRX:=XMIN;
    CURRY:=YMIN;

    IF AMPLADA < MAX_RES THEN RETURN 0;
    END IF;

    CONT=1;
    WHILE CURRY < YMAX LOOP
    WHILE CURRX < XMAX LOOP
        GEOMETRIA:=
        ST_GeomFromText('MULTIPOLYGON((( '||CURRX||'
        '||CURRY||', '||(CURRX+AMPLADA)||' '||CURRY||',
        '||(CURRX+AMPLADA)||' '||(CURRY+AMPLADA)||',
```

```

        '||CURRX||' '||(CURRY+AMPLADA)||', '||CURRX||'
        '||CURRY||'))', 3035);
    IF AMPLADA = MAX_RES THEN
        INSERT INTO TMP_PROVA_DIVISIO VALUES
            (GRD_FIXID, CONT, LEVEL+1, GEOMETRIA);

    END IF;

    PERFORM * FROM
        PUBLIC.DIVIDE (GID, GRD_FIXID, GRD_FLOAID, GRD_NEWID, CURR
            X, CURRY, MAX_RES, AMPLADA, LEVEL+1);
    CURRX:=CURRX+AMPLADA;
    CONT:=CONT+1;
    CONTOT:=CONTOT+1;

END LOOP;

    CURRY:=CURRY+AMPLADA;
    CURRX:=XMIN;
END LOOP;

RETURN CONTOT;
END;$BODY$
LANGUAGE PLPGSQL VOLATILE
COST 100;
ALTER FUNCTION PUBLIC.DIVIDE(INTEGER, CHARACTER VARYING, CHARACTER VARYING,
    CHARACTER VARYING, DOUBLE PRECISION, DOUBLE PRECISION, DOUBLE PRECISION,
    DOUBLE PRECISION, INTEGER)
OWNER TO POSTGRES;

```

Script de SQL 5Funció per obtenir la coordenada x

```
CREATE OR REPLACE FUNCTION PUBLIC.GETX_FROM_GRD_FIXID(CHARACTER VARYING)
RETURNS DOUBLE PRECISION AS
$BODY$DECLARE
    GRD_FIXID ALIAS FOR $1;
    TMP_STR CHARACTER VARYING;

BEGIN
    TMP_STR:=SUBSTRING (GRD_FIXID FROM POSITION('.' IN GRD_FIXID)+1 FOR
LENGTH (GRD_FIXID)) ;
    TMP_STR:=SUBSTRING (TMP_STR FROM POSITION('.') IN TMP_STR)+1 FOR
LENGTH (TMP_STR)) ;
    TMP_STR:=SUBSTRING (TMP_STR FROM 1 FOR POSITION('.') IN TMP_STR)-1 ) ;

RETURN CAST (TMP_STR AS DOUBLE PRECISION)/10.;
END;$BODY$
LANGUAGE PLPGSQL VOLATILE
COST 100;
ALTER FUNCTION PUBLIC.GETX_FROM_GRD_FIXID(CHARACTER VARYING)
OWNER TO POSTGRES;
```

Script de SQL 6Funció per obtenir la coordenada y

```
CREATE OR REPLACE FUNCTION PUBLIC.GETY_FROM_GRD_FIXID (CHARACTER VARYING)
RETURNS DOUBLE PRECISION AS
$BODY$DECLARE
    GRD_FIXID ALIAS FOR $1;

    TMP_STR CHARACTER VARYING;

BEGIN
    TMP_STR:=SUBSTRING (GRD_FIXID FROM POSITION('.') IN GRD_FIXID)+1 FOR
    LENGTH (GRD_FIXID) ) ;
    TMP_STR:=SUBSTRING (TMP_STR FROM POSITION('.') IN TMP_STR)+1 FOR
    LENGTH (TMP_STR) ) ;
    TMP_STR:=SUBSTRING (TMP_STR FROMPOSITION('.') IN TMP_STR)+1 FOR
    LENGTH (TMP_STR) ) ;

    TMP_STR:=SUBSTRING (TMP_STR FROM 1 FOR  LENGTH (TMP_STR) ) ;

RETURN CAST (TMP_STR AS DOUBLE PRECISION) /10.;
END;$BODY$
LANGUAGE PLPGSQL VOLATILE
COST 100;
ALTER FUNCTION PUBLIC.GETY_FROM_GRD_FIXID (CHARACTER VARYING)
OWNER TO POSTGRES;
```


Script de SQL 7 Execució de les funcions DIVIDE, GETX_FROM_GRD_FIXID i GETY_FROM_GRD_FIXID

```
SELECT
DIVIDE(0, GRD_FIXID, '', '', GETX_FROM_GRD_FIXID(GRD_FIXID), GETY_FROM_GRD_FIXID(GR
D_FIXID), 62.5, 1000, LEVEL)
FROM GRID_ETRS89_LALEA_MULTIRESOL
WHERE LEVEL=0
```

Script de SQL 8 Intersecció dels quadrats de 62.5m amb el límit de Catalunya

```
UPDATE TMP_PROVA_DIVISIO A SET GEOM=(  
SELECT ST_MULTI(ST_COLLECTIONEXTRACT(ST_INTERSECTION(A.GEOM, B.GEOM), 3))  
FROM GRID_ETRS89_LALEA_MULTITRESOL B  
WHERE A.GRD_FIXID = B.GRD_FIXID AND B.LEVEL = 0)
```

Script de SQL 9 Eliminació de la geometria de 62.5m que està fora del límit de Catalunya

```
DELETFROM TMP_PROVA_DIVISIO WHERE ST_AREA (GEOM) =0
```

Script de SQL 10Inserció de la nova geometria a la taula de la geometria

```
INSERT INTO GRID_ETRS89_LALEA_MULTIRESOL (GRD_FIXID,ORDRE_DIV,LEVEL,GEOM)  
(SELECT A.GRD_FIXID,A.ORDRE_DIV,A.LEVEL,A.GEOM FROM TMP_PROVA_DIVISIO A)
```

Script de SQL 11 Unió de la taula de la geometria amb la taula on hi ha la relació recursiva

```
INSERT INTO GRID_PARES_FILLS (ID_FILL,LEVEL_FILL,ID_PARE,LEVEL_PARE)
SELECT  A.ID,A.LEVEL,B.ID,B.LEVEL
FROM GRID_ETRS89_LALEA_MULTIRESOL A, GRID_ETRS89_LALEA_MULTIRESOL B
WHERE
A.LEVEL=4 AND B.LEVEL=3 AND
ST_INTERSECTS (A.GEOM,B.GEOM)=TRUE
AND
ST_INTERSECTION (A.GEOM,B.GEOM)=A.GEOM
```

Script de SQL 12Comprovació de la recursivitat en el nivell 4

```
SELECT * FROM GRID_ETRS89_LALEA_MULTIRESOL WHERE LEVEL=4 AND NOT EXISTS  
(SELECT 1 FROM GRID_PARES_FILLS WHERE ID_FILL=ID AND LEVEL_FILL=4 AND  
LEVEL_PARE=3)
```

Script de SQL 13Canvi de la geometria original a la geometria (ETRS89_UTM31N)

```
UPDATE GRID_ETRS89_LALEA_MULTIREVOL SET  
GEOM_UTM=ST_SETSRID(ST_TRANSFORM(GEOM,25831),25831)
```

Script de SQL 14Funció CREA_TAULA_AGREGATS

```
CREATE OR REPLACE FUNCTION PUBLIC.CREA_TAULA_AGREGATS(T_NAME TEXT)
RETURNS VOID AS
$BODY$BEGIN
EXECUTE FORMAT('
    CREATE TABLE IF NOT EXISTS %I (
        ID INTEGER,
        VALOR INTEGER
    )', 'AGG_' || T_NAME);
EXECUTE FORMAT('CREATE UNIQUE INDEX PK_ID_AGG ON %I (ID) TABLESPACE
INDEXOS', 'AGG_' || T_NAME);
END
$BODY$
LANGUAGE PLPGSQL VOLATILE
COST 100;
ALTER FUNCTION PUBLIC.CREA_TAULA_AGREGATS(TEXT)
OWNER TO POSTGRES;
```


Script de SQL 15 Funció OMPLE_AGG_MAX_RESOLUCIO

```
REATE OR REPLACE FUNCTION PUBLIC.OMPLE_AGG_MAX_RESOLUCIO(  
    T_NAME TEXT,  
    LEVEL_MAX INTEGER,  
    LEVEL_MIN INTEGER)  
    RETURNS VOID AS  
$BODY$  
BEGIN  
EXECUTE 'INSERT INTO ' || QUOTE_IDENT('AGG_' || T_NAME) || ' SELECT ID, COUNT(*)  
FROM GRID_ETRS89_LALEA_MULTIRESOL , ' || QUOTE_IDENT(T_NAME) || ' WHERE ST_WITHIN  
(POSICIO,GEOM_UTM) AND LEVEL=' || LEVEL_MAX || ' GROUP BY ID';  
END  
$BODY$  
LANGUAGE PLPGSQL VOLATILE  
    COST 100;  
ALTER FUNCTION PUBLIC.OMPLE_AGG_MAX_RESOLUCIO(TEXT, INTEGER, INTEGER)  
OWNER TO POSTGRES;
```

Script de SQL 16Funció AGREGA_EN_JERARQUIA

```
CREATE OR REPLACE FUNCTION PUBLIC.AGREGA_EN_JERARQUIA (  
    T_NAME TEXT,  
    LEVEL_MAX INTEGER,  
    LEVEL_MIN INTEGER)  
RETURNS VOID AS  
$BODY$  
DECLARE  
CURRENT_LEVEL INTEGER;  
BEGIN  
CURRENT_LEVEL=LEVEL_MAX;  
IF (CURRENT_LEVEL = LEVEL_MIN) THEN  
RETURN;  
END IF ;  
EXECUTE 'INSERT INTO ' || QUOTE_IDENT('AGG_' || T_NAME) || ' SELECT  
ID_PARE,SUM(VALOR) FROM ' || QUOTE_IDENT('AGG_' || T_NAME) || ', GRID_PARES_FILLS  
WHERE ID=ID_FILL AND LEVEL_FILL=' || CURRENT_LEVEL || ' GROUP BY ID_PARE';  
CURRENT_LEVEL:=CURRENT_LEVEL-1;  
  
PERFORM * FROM AGREGA_EN_JERARQUIA (T_NAME,CURRENT_LEVEL,LEVEL_MIN);  
END  
$BODY$  
LANGUAGE PLPGSQL VOLATILE  
COST 100;  
ALTER FUNCTION PUBLIC.AGREGA_EN_JERARQUIA(TEXT, INTEGER, INTEGER)  
OWNER TO POSTGRES;
```

Script de SQL 17 Funció CREACIO_QUADTREE_INFO

```
CREATE OR REPLACE FUNCTION PUBLIC.CREACIO_QUADTREE_INFO(  
    T_NAME TEXT,  
    LEVEL_MAX INTEGER,  
    LEVEL_MIN INTEGER)  
RETURNS VOID AS  
$BODY$BEGIN  
PERFORM * FROM CREA_Taula_Agregats(T_NAME);  
PERFORM * FROM OMPLE_AGG_MAX_RESOLUCIO(T_NAME ,LEVEL_MAX ,LEVEL_MIN );  
PERFORM * FROM AGREGA_EN_JERARQUIA(T_NAME,LEVEL_MAX,LEVEL_MIN);  
END$BODY$  
LANGUAGE PLPGSQL VOLATILE  
COST 100;  
ALTER FUNCTION PUBLIC.CREACIO_QUADTREE_INFO(TEXT, INTEGER, INTEGER)  
OWNER TO POSTGRES;
```

Script de SQL 18Funció CREA_TAULA_QUADTREE

```
CREATE OR REPLACE FUNCTION PUBLIC.CREA_TAULA_QUADTREE(  
LEVEL_MAX INTEGER,  
LEVEL_MIN INTEGER,  
T_NAME TEXT,  
LLINDAR INTEGER)  
RETURNS VOID AS  
$BODY$  
BEGIN  
EXECUTE FORMAT('CREATE TABLE IF NOT EXISTS %I (      ID INTEGER )',  
'QUADTREE_'||LEVEL_MAX||'_'||LEVEL_MIN||'_'||T_NAME||'_'||LLINDAR);  
EXECUTE FORMAT('CREATE UNIQUE INDEX  
PK_QUADTREE_'||LEVEL_MAX||'_'||LEVEL_MIN||'_'||T_NAME||'_'||LLINDAR||'  
ON %I (ID) TABLESPACE  
INDEXOS','QUADTREE_'||LEVEL_MAX||'_'||LEVEL_MIN||'_'||T_NAME||'_'||LLINDAR);  
END  
$BODY$  
LANGUAGE PLPGSQL VOLATILE  
COST 100;  
ALTER FUNCTION PUBLIC.CREA_TAULA_QUADTREE(INTEGER, INTEGER, TEXT, INTEGER)  
OWNER TO POSTGRES;
```

Script de SQL 19 Funció ANALIZE_ELEMENT

```
CREATE OR REPLACE FUNCTION PUBLIC.ANALIZE_ELEMENT (
    ID_CERCA INTEGER,
    LLINDAR INTEGER,
    TARGET TEXT)
RETURNS VOID AS
$BODY$
DECLARE
CURS REFCURSOR;
RESULT RECORD;
V INTEGER;
T INTEGER;
CAL_AGREGAR INTEGER;
IDENTIFICADORS INTEGER[];
BEGIN
    V:=0;

    CAL_AGREGAR:=MUSTAGGREGATE(ID_CERCA,LLINDAR);

    IF CAL_AGREGAR = 0 THEN
        OPEN CURS FOR SELECT ID_FILL FROM GRID_PARES_FILLS WHERE ID_PARE=ID_CERCA
            AND EXISTS (SELECT 1 FROM AGG_HABITANTS_EDAT_POSICIO WHERE
                GRID_PARES_FILLS.ID_FILL= AGG_HABITANTS_EDAT_POSICIO.ID) ;
        LOOP
            FETCH CURS INTO RESULT;
            EXIT WHEN NOT FOUND;
            V:=V+1;
            T:=RESULT.ID_FILL;
            IDENTIFICADORS[V]=T;

        END LOOP;
        CLOSE CURS;

        IF V=0 THEN

            PERFORM * FROM INSERTA_ELEMENT(ID_CERCA,TARGET);
        ELSE
            FOR I IN 1..V LOOP

                PERFORM * FROM ANALIZE_ELEMENT(IDENTIFICADORS[I],LLINDAR,TARGET);
            END LOOP;

        END IF;

    ELSE
        PERFORM * FROM INSERTA_ELEMENT(ID_CERCA,TARGET);
    END IF;
    RETURN;
END
$BODY$
LANGUAGE PLPGSQL VOLATILE
COST 100;
ALTER FUNCTION PUBLIC.ANALIZE_ELEMENT(INTEGER, INTEGER, TEXT)
OWNER TO POSTGRES;
```

Script de SQL 20 Funció MUSTAGGREGATE

```
CREATE OR REPLACE FUNCTION PUBLIC.MUSTAGGREGATE (
    ID_CERCA INTEGER,
    LLINDAR INTEGER)
RETURNS INTEGER AS
$BODY$DECLARE
    CURS REFCURSOR;
    RESULT RECORD;
    V INTEGER;
    T INTEGER;
BEGIN
    V:=0;

    OPEN CURS FOR SELECT B.VALOR,A.ID_PARE,A.ID_FILL FROM
    GRID_PARES_FILLS A,AGG_HABITANTS_EDAT_POSICIO B WHERE A.ID_FILL=B.ID AND
    A.ID_PARE=ID_CERCA ;
    LOOP
        FETCH CURS INTO RESULT;
    EXIT WHEN NOT FOUND;
        T:=RESULT.VALOR;
    IF T < LLINDAR THEN
        V:=-1;
    EXIT;
        END IF;
    END LOOP;
CLOSE CURS;
    RETURN V;

END$BODY$
LANGUAGE PLPGSQL VOLATILE
COST 100;
ALTER FUNCTION PUBLIC.MUSTAGGREGATE (INTEGER, INTEGER)
OWNER TO POSTGRES;
```

Script de SQL 21 Funció INSERTA_ELEMENT

```
CREATE OR REPLACE FUNCTION PUBLIC.INSERTA_ELEMENT(  
    ID_INSERTAR INTEGER,  
    TARGET TEXT)  
RETURNS VOID AS  
$BODY$BEGIN  
  
EXECUTE 'INSERT INTO ' ||QUOTE_IDENT(TARGET) || ' (ID)  
VALUES(' || ID_INSERTAR || ')';  
  
END $BODY$  
LANGUAGE PLPGSQL VOLATILE  
COST 100;  
ALTER FUNCTION PUBLIC.INSERTA_ELEMENT(INTEGER, TEXT)  
OWNER TO POSTGRES;
```

Script de SQL 22 Funció CREA_QUADTREE

```
CREATE OR REPLACE FUNCTION PUBLIC.CREA_QUADTREE (
    LEVEL_MAX INTEGER,
    LEVEL_MIN INTEGER,
    T_NAME TEXT,
    LLINDAR INTEGER)
RETURNS VOID AS
$BODY$DECLARE
    TARGET TEXT;
    CURS REFCURSOR;
    RESULT RECORD;

    T INTEGER;
    EXPRESIO_SQL TEXT;
BEGIN

    PERFORM * FROM CREA_TAUULA_QUADTREE (LEVEL_MAX, LEVEL_MIN, T_NAME, LLINDAR);

    TARGET:='QUADTREE_' || LEVEL_MAX || '_' || LEVEL_MIN || '_' || T_NAME || '_' || LLINDAR; FOR
    RESULT IN SELECT B.ID AS IDENTIFICADOR FROM GRID_ETRS89_LALEA_MULTIRESOL B
    WHERE B.LEVEL=LEVEL_MIN AND EXISTS (SELECT 1 FROM QUOTE_IDENT('AGG_' ||
    T_NAME) WHERE B.ID=ID)
    LOOP
        T:=RESULT.IDENTIFICADOR;
        PERFORM * FROM ANALIZE_ELEMENT (T, LLINDAR, TARGET);

    ENDLLOOP;
END $BODY$
LANGUAGE PLPGSQL VOLATILE
COST 100;
ALTER FUNCTION PUBLIC.CREA_QUADTREE (INTEGER, INTEGER, TEXT, INTEGER)
OWNER TO POSTGRES;
```


Script de SQL 23 Reunió de la informació agregada de la població i de la geometria de cada element del quadtree, en el cas del quadtree QT_4_3_HABITANTS_EDAT_POSICIO_13

```
CREATE TABLE QUADTREEG_4_3_HABITANTS_EDAT_POSICIO_13 AS
SELECT C.ID AS ID,B.LEVEL AS LEVEL,A.VALOR AS POB,B.GEOM_UTM AS GEOM FROM
AGG_HABITANTS_EDAT_POSICIO A,GRID_ETRS89_LALEA_MULTIRESOL
B,QUADTREE_4_3_HABITANTS_EDAT_POSICIO_13 C
WHERE A.ID=B.ID AND C.ID=B.ID
```

Script de SQL 24Funció de creació de l'espai de comparació

```
-- FUNCTION: PUBLIC.CODIFICA_ESPAI_COMPARACIO(TEXT, NUMERIC, TEXT, TEXT)
-- DROP FUNCTION PUBLIC.CODIFICA_ESPAI_COMPARACIO(TEXT, NUMERIC, TEXT, TEXT);
CREATE OR REPLACE FUNCTION PUBLIC.CODIFICA_ESPAI_COMPARACIO(
SEXO TEXT,
EDAT NUMERIC,
NACI TEXT,
CPRON TEXT)

RETURNS INTEGER[] AS
$BODY$DECLARE
SEXO ALIAS FOR $1;
EDAT ALIAS FOR $2;
NACI ALIAS FOR $3;
CPRON ALIAS FOR $4;
/*
RECODIFICA UN HABITANT EN L'ESPAI AGREGAT DE VARIABLES
*/
RECODE INTEGER[];
BEGIN

/* SEXE */
    IF SEXO = '1' THEN RECODE[0]=1; END IF;
    IF SEXO = '6' THEN RECODE[0]=2; END IF;

/* EDAT */
    IF EDAT >= 0 AND EDAT < 10 THEN RECODE[1]=1; END IF;
    IF EDAT >= 10 AND EDAT < 20 THEN RECODE[1]=2; END IF;
    IF EDAT >= 20 AND EDAT < 30 THEN RECODE[1]=3; END IF;
    IF EDAT >= 30 AND EDAT < 40 THEN RECODE[1]=4; END IF;
    IF EDAT >= 40 AND EDAT < 50 THEN RECODE[1]=5; END IF;
    IF EDAT >= 50 AND EDAT < 60 THEN RECODE[1]=6; END IF;
    IF EDAT >= 60 AND EDAT < 70 THEN RECODE[1]=7; END IF;
    IF EDAT >= 70 AND EDAT < 80 THEN RECODE[1]=8; END IF;
    IF EDAT >= 80 AND EDAT < 90 THEN RECODE[1]=9; END IF;
    IF EDAT >=90 THEN RECODE[1]=10; END IF;
/* NACIONALITAT */
    IF NACI = '108' THEN RECODE[2]=1;
    ELSE RECODE[2]=2;
    END IF;
/* LLOC NAIXEMENT */
    RECODE[3]=2;
    IF CPRON='66' THEN RECODE[3]=3; END IF;
    IF CPRON = '08' OR CPRON='17' OR CPRON='25' OR CPRON='43' THEN RECODE[3]=1;
END IF;

RETURN RECODE;

END;$BODY$
LANGUAGE PLPGSQL VOLATILE
COST 100;

ALTER FUNCTION PUBLIC.CODIFICA_ESPAI_COMPARACIO(TEXT, NUMERIC, TEXT, TEXT)
OWNER TO POSTGRES;
```

Script de SQL 25Funció de creació de l'espai de publicació

```
- FUNCTION: PUBLIC.CODIFICA_ESPAI_PUBLICACIO(TEXT, NUMERIC, TEXT, TEXT)

-- DROP FUNCTION PUBLIC.CODIFICA_ESPAI_PUBLICACIO(TEXT, NUMERIC, TEXT, TEXT);

CREATE OR REPLACE FUNCTION PUBLIC.CODIFICA_ESPAI_PUBLICACIO(
    SEXO TEXT,
    EDAT NUMERIC,
    NACI TEXT,
    CPRON TEXT)
    RETURNS INTEGER[] AS
$BODY$DECLARE
    SEXO ALIAS FOR $1;
    EDAT ALIAS FOR $2;
    NACI ALIAS FOR $3;
    CPRON ALIAS FOR $4;

/*
RECODIFICA UN HABITANT EN L'ESPAI AGREGAT DE VARIABLES
*/
    RECODE INTEGER[];

BEGIN

/* SEXE */
    IF SEXO = '1' THEN RECODE[0]=1; END IF;
    IF SEXO = '6' THEN RECODE[0]=2; END IF;

/* EDAT */
    IF EDAT >= 0 AND EDAT < 15 THEN RECODE[1]=1; END IF;
    IF EDAT >= 15 AND EDAT < 30 THEN RECODE[1]=2; END IF;
    IF EDAT >= 30 AND EDAT < 45 THEN RECODE[1]=3; END IF;
    IF EDAT >= 45 AND EDAT < 60 THEN RECODE[1]=4; END IF;
    IF EDAT >= 60 AND EDAT < 75 THEN RECODE[1]=5; END IF;
    IF EDAT >= 75 AND EDAT < 90 THEN RECODE[1]=6; END IF;
    IF EDAT >= 90 THEN RECODE[1]=19; END IF;

/* NACIONALITAT */
    IF NACI = '108' THEN RECODE[2]=1;
    ELSE RECODE[2]=2;
    END IF;

/* LLOC NAIXEMENT */
    RECODE[3]=2;
    IF CPRON='66' THEN RECODE[3]=3; END IF;
    IF CPRON = '08' OR CPRON='17' OR CPRON='25' OR CPRON='43' THEN RECODE[3]=1;
END IF;

    RETURN RECODE;

END;$BODY$
LANGUAGE PLPGSQL VOLATILE
COST 100;
ALTER FUNCTION PUBLIC.CODIFICA_ESPAI_PUBLICACIO(TEXT, NUMERIC, TEXT, TEXT)
    OWNER TO POSTGRES;
```

Script de SQL 26 Creació de la taula dels experiments de Montecarlo

```
CREATE TABLE PUBLIC.RESULTAT_EXPERIMENT_MONTECARLO
(
  EXPERIMENT INTEGER,
  POB_EXACTE DOUBLE PRECISION,
  POB_4_3_13 DOUBLE PRECISION,
  POB_4_2_13 DOUBLE PRECISION,
  POB_4_1_13 DOUBLE PRECISION,
  POB_4_4_17 DOUBLE PRECISION,
  POB_4_3_17 DOUBLE PRECISION,
  POB_4_2_17 DOUBLE PRECISION,
  POB_4_1_17 DOUBLE PRECISION,
  POB_4_3_19 DOUBLE PRECISION,
  POB_4_2_19 DOUBLE PRECISION,
  POB_4_1_19 DOUBLE PRECISION,
  POB_4_3_23 DOUBLE PRECISION,
  POB_4_2_23 DOUBLE PRECISION,
  POB_4_1_23 DOUBLE PRECISION
)
WITH (
  OIDS=FALSE
)
TABLESPACE DADES;
ALTER TABLE PUBLIC.RESULTAT_EXPERIMENT_MONTECARLO
OWNER TO POSTGRES;
-- INDEX: PUBLIC.IS_EXPERIMENT
-- DROP INDEX PUBLIC.IS_EXPERIMENT;
CREATE UNIQUE INDEX IS_EXPERIMENT
ON PUBLIC.RESULTAT_EXPERIMENT_MONTECARLO
USING BTREE
  (EXPERIMENT)
TABLESPACE INDEXOS;
```

Script de SQL 27 Càlcul de la població exacta ens els polígons de Montecarlo

```
INSERT INTO RESULTAT_EXPERIMENT_MONTECARLO (EXPERIMENT, POB_EXACTE)
SELECT EXPERIMENT, COUNT (*)
FROM MONTECARLO_EXPERIMENTS_POLYGONS, HABITANTS_EDAT_POSICIO WHERE
ST_INTERSECTS (POSICIO, GEOM)
GROUP BY EXPERIMENT
```

Script de SQL 28 Càlcul de la població segons cada quadtree

```
UPDATE RESULTAT_EXPERIMENT_MONTECARLO SET POB_4_3_13=
(SELECT SUM(B.POB*(ST_AREA(ST_INTERSECTION(A.GEOM,B.GEOM))/ST_AREA(B.GEOM)))
FROM
MONTECARLO_EXPERIMENTS_POLYGONS A,QUADTREEG_4_3_HABITANTS_EDAT_POSICIO_13 B
WHERE ST_INTERSECTS(A.GEOM,B.GEOM)
AND RESULTAT_EXPERIMENT_MONTECARLO.EXPERIMENT=A.EXPERIMENT)
```

Script de SQL 29 Càlcul de l'error relatiu en els experiments

```
(SELECT ABS(POB_EXACTE-POB_4_3_23)/POB_EXACTE AS ERR_4_3_23,  
ABS(POB_EXACTE-POB_4_2_23)/POB_EXACTE AS ERR_4_2_23,  
ABS(POB_EXACTE-POB_4_1_23)/POB_EXACTE AS ERR_4_1_23  
FROM RESULTAT_EXPERIMENT_MONTECARLO)  
TO'D:/ANALISI/ERR_LLINDAR_23.CSV' CSV HEADER DELIMITER ','
```

Script de SQL 30Càlcul de les diferències relatives

```
SELECT A.EXPERIMENT,A.GEOM,
(POB_4_3_13-POB_EXACTE)/POB_EXACTE AS D_4_3_13,
(POB_4_2_13-POB_EXACTE)/POB_EXACTE AS D_4_2_13,
(POB_4_1_13-POB_EXACTE)/POB_EXACTE AS D_4_1_13,
(POB_4_4_17-POB_EXACTE)/POB_EXACTE AS D_4_4_17,
(POB_4_3_17-POB_EXACTE)/POB_EXACTE AS D_4_3_17,
(POB_4_2_17-POB_EXACTE)/POB_EXACTE AS D_4_2_17,
(POB_4_1_17-POB_EXACTE)/POB_EXACTE AS D_4_1_17,
(POB_4_3_19-POB_EXACTE)/POB_EXACTE AS D_4_3_19,
(POB_4_2_19-POB_EXACTE)/POB_EXACTE AS D_4_2_19,
(POB_4_1_19-POB_EXACTE)/POB_EXACTE AS D_4_1_19,
(POB_4_3_23-POB_EXACTE)/POB_EXACTE AS D_4_3_23,
(POB_4_2_23-POB_EXACTE)/POB_EXACTE AS D_4_2_23,
(POB_4_1_23-POB_EXACTE)/POB_EXACTE AS D_4_1_23

FROM MONTECARLO_EXPERIMENTS_POLYGONS A,RESULTAT_EXPERIMENT_MONTECARLO B
WHERE A.EXPERIMENT=B.EXPERIMENT

ABS (POB_EXACTE-POB_4_3_23)/POB_EXACTE AS ERR_4_3_23
```



```
if edat >= 0 and edat < 5 then recode[1]=1; endif;  
if edat >= 5 and edat < 10 then recode[1]=2; endif;  
if edat >= 10 and edat < 15 then recode[1]=3; endif;  
if edat >= 15 and edat < 20 then recode[1]=4; endif;  
if edat >= 20 and edat < 25 then recode[1]=5; endif;  
if edat >= 25 and edat < 30 then recode[1]=6; endif;  
if edat >= 30 and edat < 35 then recode[1]=7; endif;  
if edat >= 35 and edat < 40 then recode[1]=8; endif;  
if edat >= 40 and edat < 45 then recode[1]=9; endif;  
if edat >= 45 and edat < 50 then recode[1]=10; endif;  
if edat >= 50 and edat < 55 then recode[1]=11; endif;  
if edat >= 55 and edat < 60 then recode[1]=12; endif;  
if edat >= 60 and edat < 65 then recode[1]=13; endif;  
if edat >= 65 and edat < 70 then recode[1]=14; endif;  
if edat >= 70 and edat < 75 then recode[1]=15; endif;  
if edat >= 75 and edat < 80 then recode[1]=16; endif;  
if edat >= 80 and edat < 85 then recode[1]=17; endif;  
if edat >= 85 and edat < 90 then recode[1]=18; endif;  
if edat >= 90 then recode[1]=19; endif;
```

Script de Java 1 Fórmula de la funció de perturbació de coordenades

MAIN

```
PACKAGE ORG.IDESCAT.PERTORBACIO;
/*
 * TO CHANGE THIS LICENSE HEADER, CHOOSE LICENSE HEADERS IN PROJECT
PROPERTIES.
 * TO CHANGE THIS TEMPLATE FILE, CHOOSE TOOLS | TEMPLATES
 * AND OPEN THE TEMPLATE IN THE EDITOR.
 */
/**
 *
 * @AUTHOR
 */
IMPORT JAVA.SQL.*;
PUBLIC CLASS MAIN {
    /**
     * @PARAMARGS THE COMMAND LINE ARGUMENTS
     */
    PUBLIC STATIC VOID MAIN(String[] args) THROWS EXCEPTION{
        // TODO CODE APPLICATION LOGIC HERE
        THROW NEW EXCEPTION ();
        SYSTEM.OUT.PRINTLN("HOLAMUNDO");
        JAVA.SQL.CONNECTION C=NULL;
        CONNECTION CWRITE=NULL;
        TRY
        {

CLASS.FORNAME("ORG.POSTGRESQL.DRIVER");
        // READER
        C= JAVA.SQL.DRIVERMANAGER.GETCONNECTION();
        CWRITE= JAVA.SQL.DRIVERMANAGER.GETCONNECTION("");
        LECTORHABITANTS LH=NEW LECTORHABITANTS(C,CWRITE);
        LH.LLEGEIX();
        }
        CATCH(EXCEPTION E){E.PRINTSTACKTRACE(SYSTEM.OUT);}
        TRY{C.CLOSE();CWRITE.CLOSE();}CATCH(EXCEPTION U){}

    }
}
```

ESCRIPTORMATRIU JAVA

```
*
 * TO CHANGE THIS LICENSE HEADER, CHOOSE LICENSE HEADERS IN PROJECT
PROPERTIES.
 * TO CHANGE THIS TEMPLATE FILE, CHOOSE TOOLS | TEMPLATES
 * AND OPEN THE TEMPLATE IN THE EDITOR.
 */
PACKAGE ORG.IDESCAT.PERTORBACIO;

/**
 *
 * @AUTHOR
 */
IMPORT JAVA.SQL.*;
PUBLIC CLASS ESCRIPTOR_MATRIU {
    CONNECTION C=NULL;
    STRING INSERT="INSERT INTO MATRIU_INTERCANVI
(GID_1,GID_2,INVERS_DISTANCIA) "
+" SELECT A.GID,B.GID,1./ST_DISTANCE(A.POSICIO,B.POSICIO)
FROMHABITANTS_EN_ESPAIS_CLASIFICACIO A, HABITANTS_EN_ESPAIS_CLASIFICACIO B"
+" WHERE A.GID=? AND A.GID<>B.GID AND A.ID_COMPARACIO=B.ID_COMPARACIO AND
A.ID_PUBLICACIO<>B.ID_PUBLICACIO AND ST_DISTANCE(A.POSICIO,B.POSICIO) > 0
ORDER BY ST_DISTANCE(A.POSICIO,B.POSICIO) LIMIT 25";
    PREPAREDSTATEMENT P;
    INTCONTADOR=0;
    LONG T_0;
    PUBLIC ESCRIPTOR_MATRIU(CONNECTION C) THROWS EXCEPTION
```

```

        {
        THIS.C=C;
        PS=C.PREPARESTATEMENT(INSERT);
            T_0=SYSTEM.CURRENTTIMEMILLIS();
        }

        PUBLIC VOID ESCRIU(INTGID,INTCONTA) THROWS EXCEPTION
        {
        PS.CLEARPARAMETERS();
        PS.SETINT(1, GID);

        CONTADOR+= PS.EXECUTEUPDATE();

        IF( (GID % 50) == 0)
        {
            LONG T_1=SYSTEM.CURRENTTIMEMILLIS();
            SYSTEM.OUT.PRINTLN("LLEGITS..... "+CONTA+" ESCRITS .... "+CONTADOR+" TEMPS
            "+(T_1-T_0)/1000);
        }
    }
}

```

LACTOR HABITANTS JAVA

```

*
* TO CHANGE THIS LICENSE HEADER, CHOOSE LICENSE HEADERS IN PROJECT
PROPERTIES.
* TO CHANGE THIS TEMPLATE FILE, CHOOSE TOOLS | TEMPLATES
* AND OPEN THE TEMPLATE IN THE EDITOR.
*/
PACKAGE ORG.IDESCAT.PERTORBACIO;
IMPORT JAVA.SQL.*;

/**
 *
 * @AUTHOR
 */
PUBLIC CLASS LECTORHABITANTS {

    CONNECTION C;
    CONNECTION CWRITE;
    STRING QUERY="SELECT GID FROM HABITANTS_EN_ESPAIS_CLASIFICACIO";
    ESCRIPTOR_MATRIU M;

    PUBLIC LECTORHABITANTS(CONNECTION C,CONNECTIONCWRITE) THROWS EXCEPTION
    {
    THIS.C=C;
    THIS.CWRITE=CWRITE;
        M=NEW ESCRIPTOR_MATRIU(CWRITE);
    }

    PUBLIC VOID LLEGEIX() THROWS EXCEPTION
    {
        STATEMENT S= C.CREATESTATEMENT();

        RESULTSETRS=S.EXECUTEQUERY(QUERY);
        INT J=0;
        WHILE(RS.NEXT())
        {
            INTGID=RS.GETINT(1);

            M.ESCRIU(GID,J++);
        }
    }
}

```

Script de Java 2 F rmula de creaci  de la matriu d'intercanvi

MAIN

```
PACKAGE ORG.IDESCAT.PERTORBACIO;

/*
 * TO CHANGE THIS LICENSE HEADER, CHOOSE LICENSE HEADERS IN PROJECT
 * PROPERTIES.
 * TO CHANGE THIS TEMPLATE FILE, CHOOSE TOOLS | TEMPLATES
 * AND OPEN THE TEMPLATE IN THE EDITOR.
 */

/**
 *
 * @AUTHOR
 */
IMPORT JAVA.SQL.*;
PUBLIC CLASS OBTEDONANTS_PAS1 {

    /**
     * @PARAMARGS THE COMMAND LINE ARGUMENTS
     */
    PUBLIC STATIC VOID MAIN(STRING[] ARGS) THROWS EXCEPTION {
        // TODO CODE APPLICATION LOGIC HERE

        SYSTEM.OUT.PRINTLN("HOLAMUNDO");
        JAVA.SQL.CONNECTION C=NULL;
        JAVA.SQL.CONNECTION C2=NULL;
        CONNECTION CWRITE=NULL;
        TRY
        {

            CLASS.FORNAME("ORG.POSTGRESQL.DRIVER");
            // READER
            C= JAVA.SQL.DRIVERMANAGER.GETCONNECTION();
            C2= JAVA.SQL.DRIVERMANAGER.GETCONNECTION();
            CWRITE= JAVA.SQL.DRIVERMANAGER.GETCONNECTION();
            LECTORPUNTSCALCULATSLH=NEW LECTORPUNTSCALCULATS(C,C2,CWRITE);
            LH.LLEGEIX();
        }
        CATCH(EXCEPTION E){E.PRINTSTACKTRACE(SYSTEM.OUT);}
        TRY{C.CLOSE();CWRITE.CLOSE();}CATCH(EXCEPTION U){}

    }

}
```

LECTOR HABITANTS

```
/*
 * TO CHANGE THIS LICENSE HEADER, CHOOSE LICENSE HEADERS IN PROJECT
 * PROPERTIES.
 * TO CHANGE THIS TEMPLATE FILE, CHOOSE TOOLS | TEMPLATES
 * AND OPEN THE TEMPLATE IN THE EDITOR.
 */
PACKAGE ORG.IDESCAT.PERTORBACIO;
IMPORT JAVA.SQL.*;

/**
 *
 * @AUTHOR
 */
PUBLIC CLASS LECTORHABITANTS {

    CONNECTION C;
    CONNECTION CWRITE;
    STRING QUERY="SELECT GID FROM HABITANTS_EN_ESPAIS_CLASIFICACIO";
    ESCRIPTOR_MATRIU M;
```

```

        PUBLIC LECTORHABITANTS (CONNECTION C, CONNECTIONCWRITE) THROWS EXCEPTION
        {
            THIS.C=C;
            THIS.CWRITE=CWRITE;
            M=NEW ESCRIPTOR_MATRIU(CWRITE);
        }

        PUBLIC VOID LLEGEIX() THROWS EXCEPTION
        {
            STATEMENT S= C.CREATESTATEMENT();

RESULTSETRS=S.EXECUTEQUERY(QUERY);
INT J=0;
            WHILE (RS.NEXT())
            {
                INTGID=RS.GETINT(1);

M.ESCRIU(GID,J++);

            }
        }
    }
}

ESCRIPTORDONANTS
/*
 * TO CHANGE THIS LICENSE HEADER, CHOOSE LICENSE HEADERS IN PROJECT
PROPERTIES.
 * TO CHANGE THIS TEMPLATE FILE, CHOOSE TOOLS | TEMPLATES
 * AND OPEN THE TEMPLATE IN THE EDITOR.
 */
PACKAGE ORG.IDESCAT.PERTORBACIO;

/**
 *
 * @AUTHOR
 */
IMPORT JAVA.SQL.*;
PUBLIC CLASS ESCRIPTORDONANTS {
    CONNECTION C=NULL;
    CONNECTION C2=NULL;
    STRING LECTURA="SELECT A.GID_2,A.INVERS_DISTANCIA FROM MATRIU_INTERCANVI A
WHERE A.GID_1=";
    STRING INSERT="INSERT INTO PUNTS_INTERCANVI VALUES (?,?)";
    PREPAREDSTATEMENTPS;
    INTCONTADOR=0;
    LONG T_0;
    PUBLIC ESCRIPTORDONANTS(CONNECTION C2,CONNECTION C) THROWS EXCEPTION
    {
        THIS.C=C;
        THIS.C2=C2;
        PS=C.PREPARESTATEMENT(INSERT);
        T_0=SYSTEM.CURRENTTIMEMILLIS();
    }

    PUBLIC VOID ESCRIU(INTGID,INTCONTA) THROWS EXCEPTION
    {
        STATEMENT S=C2.CREATESTATEMENT();
        RESULTSETRS=S.EXECUTEQUERY(LECTURA+GID);
        JAVA.UTIL.VECTOR IDS=NEW JAVA.UTIL.VECTOR();
        JAVA.UTIL.VECTORDST=NEW JAVA.UTIL.VECTOR();

        WHILE (RS.NEXT())
        {
            IDS.ADDELEMENT(RS.GETINT(1));
            DST.ADDELEMENT(RS.GETDOUBLE(2));

```

```

    }

    RS.CLOSE();
    S.CLOSE();
    JAVA.UTIL.RANDOMRND=NEW JAVA.UTIL.RANDOM(SYSTEM.CURRENTTIMEMILLIS());
    SELECCIO_DONANTMOTOR_SELECCIO=NEW SELECCIO_DONANT(GID,IDS,DST,RND);

    INTRESULTAT=MOTOR_SELECCIO.SELECCIONA_DONANT();

    PS.SETINT(1, GID);
    PS.SETINT(2, RESULTAT);

    PS.EXECUTEUPDATE();

    PS.CLEARPARAMETERS();

    }

}

SELECCIODONANT
/*
 * TO CHANGE THIS LICENSE HEADER, CHOOSE LICENSE HEADERS IN PROJECT
PROPERTIES.
 * TO CHANGE THIS TEMPLATE FILE, CHOOSE TOOLS | TEMPLATES
 * AND OPEN THE TEMPLATE IN THE EDITOR.
 */
PACKAGE ORG.IDESCAT.PERTORBACIO;

/**
 *
 * @AUTHOR
 */
IMPORT JAVA.UTIL.*;
PUBLIC CLASS SELECCIO_DONANT {
    INTID_TARGET;
    VECTOR ID_DONANTS;
    VECTOR INVERS_DISTANCIES;
    RANDOM RND;
    PUBLIC
    SELECCIO_DONANT(INTTARGET,VECTORIDS,VECTORDISTANCIES, JAVA.UTIL.RANDOMRND)
    {
        ID_TARGET=TARGET;
        ID_DONANTS=IDS;
        INVERS_DISTANCIES=DISTANCIES;
        THIS.RND=RND;
    }

    PUBLIC INTSELECCIONA_DONANT() THROWS EXCEPTION
    {
        DOUBLE SUM_DIST=0.;

        FOR(INTI=0; I<INVERS_DISTANCIES.SIZE(); I++)
        {
            SUM_DIST+=(DOUBLE) INVERS_DISTANCIES.ELEMENTAT(I);
        }

        VECTOR DST_NORM=NEW VECTOR();
        FOR(INTI=0; I<INVERS_DISTANCIES.SIZE(); I++)
        {
            DST_NORM.ADD((DOUBLE) INVERS_DISTANCIES.ELEMENTAT(I)/SUM_DIST);
        }

        DOUBLE INF[]= NEW DOUBLE[INVERS_DISTANCIES.SIZE()];
        DOUBLE SUP[]= NEW DOUBLE[INVERS_DISTANCIES.SIZE()];

```

```

        DOUBLE ACUM=0;
        FOR (INTI=0; I<INVERS_DISTANCIES.SIZE(); I++)
        {
            INF[I]=ACUM;
            SUP[I]=ACUM+(DOUBLE) DST_NORM.ELEMENTAT(I);

            ACUM+=(DOUBLE) DST_NORM.ELEMENTAT(I);
        }

        DOUBLE ALEA=RND.NEXTDOUBLE();
        /*
        SYSTEM.OUT.PRINTLN("ALEATORI.... "+ALEA);
        FOR (INTI=0; I<INVERS_DISTANCIES.SIZE(); I++)
        {
            SYSTEM.OUT.PRINTLN("ID.. "+ID_DONANTS.ELEMENTAT(I)+" DISTANCIA...
            "+(DOUBLE) INVERS_DISTANCIES.ELEMENTAT(I)+" INF.. "+INF[I]+" SUP.. "+SUP[I]);
        }
        */

        INTSELECCIONAT=-1;
        FOR (INTI=0; I<INVERS_DISTANCIES.SIZE(); I++)
        {
            IF (ALEA>= INF[I] && ALEA<=SUP[I]) {SELECCIONAT=I; BREAK;}
        }

        IF (SELECCIONAT == -1 ) THROW NEW EXCEPTION("MERDA...");

        RETURN (INTEGER) THIS.ID_DONANTS.ELEMENTAT (SELECCIONAT);
    }
}

```

Script de Java 3Emplenament de la taula d' HABITANTS_EDAT_POSICIO_PERTURBAT

MAIN

```
PERTURBA
/*
 * TO CHANGE THIS LICENSE HEADER, CHOOSE LICENSE HEADERS IN PROJECT
PROPERTIES.
 * TO CHANGE THIS TEMPLATE FILE, CHOOSE TOOLS | TEMPLATES
 * AND OPEN THE TEMPLATE IN THE EDITOR.
 */
PACKAGE ORG.IDESCAT.PERTORBACIO;

/**
 *
 * @AUTHOR
 */
IMPORT JAVA.SQL.*;
PUBLIC CLASS PERTURBA {

    PUBLIC STATIC VOID MAIN(STRING[] ARGS) THROWS EXCEPTION{
        // TODO CODE APPLICATION LOGIC HERE

SYSTEM.OUT.PRINTLN("HOLAMUNDO");
JAVA.SQL.CONNECTION C=NULL;
        CONNECTION C2=NULL;
        CONNECTION CWRITE=NULL;
        TRY
        {

CLASS.FORNAME("ORG.POSTGRESQL.DRIVER");
            C= JAVA.SQL.DRIVERMANAGER.GETCONNECTION();
            C2= JAVA.SQL.DRIVERMANAGER.GETCONNECTION();
CWRITE= JAVA.SQL.DRIVERMANAGER.GETCONNECTION();

            STRING SELECT_PUNTS="SELECT GID_1,GID_2 FROM PUNTS_INTERCANVI";
            STRING SELECT_PERSONA="SELECT GID, PRO,MUN,DIS,SEC,CPRON,CMUNN,
EDAT,SEXO,NACI,ID_FONT,D_QUALITAT,ST_X(POSICIO),ST_Y(POSICIO),FK_GRID250 FROM
HABITANTS_EDAT_POSICIO WHERE GID=?";

            STRING INSERT_PERSONA="INSERT INTO HABITANTS_EDAT_POSICIO_PERTURBAT
VALUES (?, ?, ?, ?, ?, ?, ?, ?, ?, ?, ?, ?, ST_SETSRID(ST_MAKEPOINT(?, ?),25831) , ?)";

            STATEMENT S=C.CREATESTATEMENT();
PREPAREDSTATEMENT P1=C2.PREPARESTATEMENT(SELECT_PERSONA);
PREPAREDSTATEMENTP_ESCRIU=CWRITE.PREPARESTATEMENT(INSERT_PERSONA);

RESULTSETRS=S.EXECUTEQUERY(SELECT_PUNTS);

            WHILE(RS.NEXT())
            {

INT GID_1=RS.GETINT(1);
INT GID_2=RS.GETINT(2);

                PERSONA PERSONA1=NEW PERSONA();
                PERSONA PERSONA2=NEW PERSONA();

                P1.CLEARPARAMETERS();
                P1.SETINT(1, GID_1);
RESULTSET RS1=P1.EXECUTEQUERY();
                PERSONA1.LEE(RS1);
                P1.CLEARPARAMETERS();
                P1.SETINT(1, GID_2);
RESULTSET RS2=P1.EXECUTEQUERY();
```



```

PERSONA2.LEE (RS2) ;

// INTERCANVIACORRDENADES..
DOUBLE TMP_X=PERSONA1.X;
DOUBLETMP_Y=PERSONA1.Y;

PERSONA1.X=PERSONA2.X;
PERSONA1.Y=PERSONA2.Y;

PERSONA2.X=TMP_X;
PERSONA2.Y=TMP_Y;

PERSONA1.ESCRIU (P_ESCRIU) ;
PERSONA2.ESCRIU (P_ESCRIU) ;

}
RS.CLOSE () ;
S.CLOSE () ;
P1.CLOSE () ;
P_ESCRIU.CLOSE () ;

}
CATCH (EXCEPTION E) {E.PRINTSTACKTRACE (SYSTEM.OUT) ; }
TRY {C.CLOSE () ; CWRITE.CLOSE () ; } CATCH (EXCEPTION U) {}

}

}

```

PERSONA

```

/*
 * TO CHANGE THIS LICENSE HEADER, CHOOSE LICENSE HEADERS IN PROJECT
PROPERTIES.
 * TO CHANGE THIS TEMPLATE FILE, CHOOSE TOOLS | TEMPLATES
 * AND OPEN THE TEMPLATE IN THE EDITOR.
 */
PACKAGE ORG.IDESCAT.PERTORBACIO;

/**
 *
 * @AUTHOR PRACTIQUES_MCASTELLO
 */
IMPORT JAVA.SQL.*;
PUBLIC CLASS PERSONA {
    PUBLIC INTGID;
    PUBLIC STRING PRO;
    PUBLIC STRING MUN;
    PUBLIC STRING DIS;
    PUBLIC STRING SEC;
    PUBLIC STRING CPRON;
    PUBLIC STRING CMUNN;
    PUBLIC DOUBLE EDAT;
    PUBLIC STRING SEXO;
    PUBLIC STRING NACI;
    PUBLIC INTID_FONT;
    PUBLIC INTD_QUALITAT;
    PUBLIC DOUBLE X;
    PUBLIC DOUBLE Y;
    PUBLIC INT FK_GRID250;

    PUBLIC VOID LEE (RESULTSETRS) THROWS EXCEPTION
    {
RS.NEXT () ;
GID=RS.GETINT (1) ;
    PRO=RS.GETSTRING (2) ;
MUN=RS.GETSTRING (3) ;

```

```

        DIS=RS.GETSTRING(4);
        SEC=RS.GETSTRING(5);
        CPRON=RS.GETSTRING(6);
        CMUNN=RS.GETSTRING(7);
        EDAT=RS.GETDOUBLE(8);
        SEXO=RS.GETSTRING(9);
        NACI=RS.GETSTRING(10);
        ID_FONT=RS.GETINT(11);
        D_QUALITAT=RS.GETINT(12);
        X=RS.GETDOUBLE(13);
        Y=RS.GETDOUBLE(14);
        FK_GRID250=RS.GETINT(15);

    }

    PUBLIC VOID ESCRIU(PREPAREDSTATEMENTPR)
    {
        TRY
        {
PR.CLEARPARAMETERS();

PR.SETINT(1, GID);
PR.SETSTRING(2, PRO);
PR.SETSTRING(3, MUN);
PR.SETSTRING(4, DIS);
PR.SETSTRING(5, SEC);
PR.SETSTRING(6, CPRON);
PR.SETSTRING(7, CMUNN);
PR.SETDOUBLE(8, EDAT);
PR.SETSTRING(9, SEXO);
PR.SETSTRING(10, NACI);
PR.SETINT(11, ID_FONT);
PR.SETINT(12, D_QUALITAT);
PR.SETDOUBLE(13,X);
PR.SETDOUBLE(14,Y);
PR.SETINT(15, THIS.FK_GRID250);

PR.EXECUTEUPDATE();
        }CATCH(EXCEPTION U){SYSTEM.OUT.PRINTLN(U.GETMESSAGE());}

    }
}

```